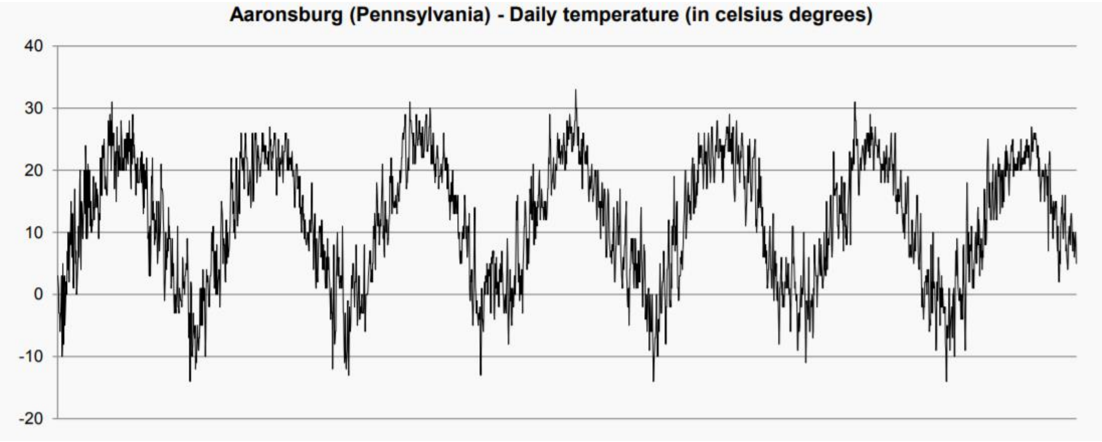
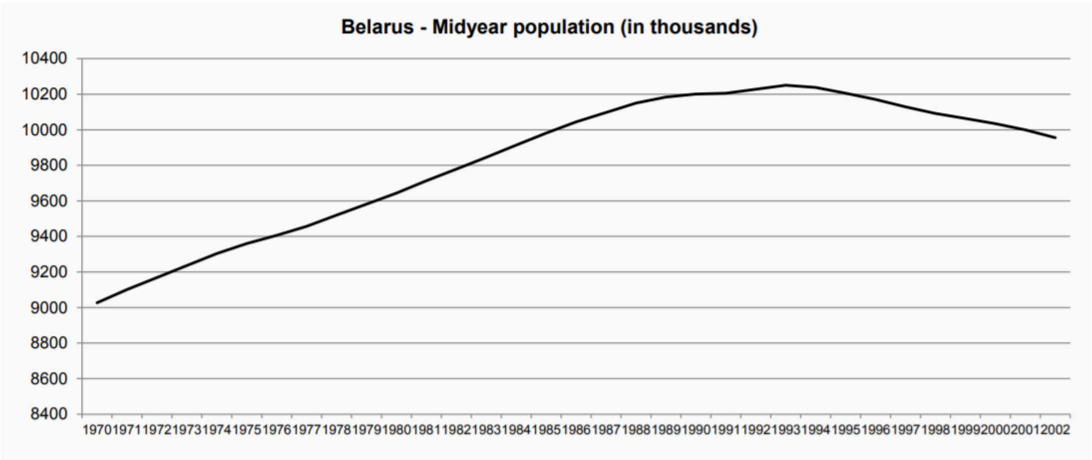
Graphical and Data Analysis using the R Software Package

Week 1: Introduction to Forecasting

UNIVERSITY of NICOSIA

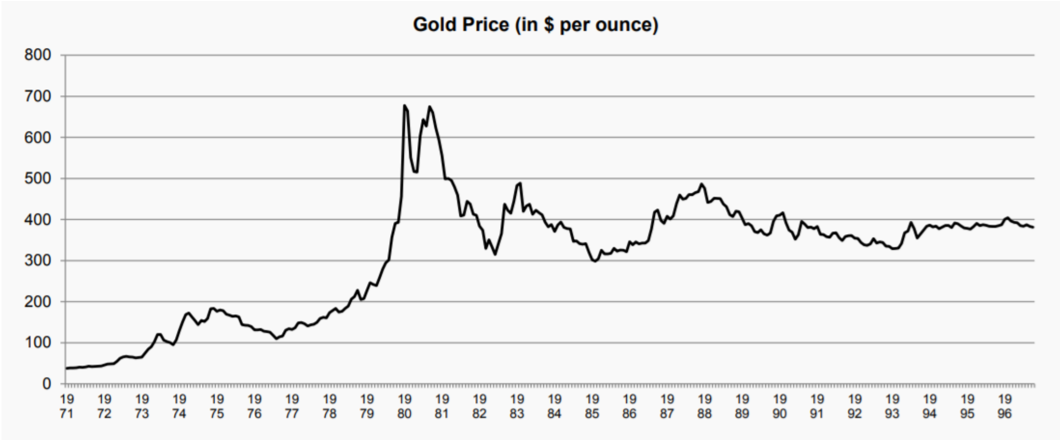
Why visualize and analyze your data?

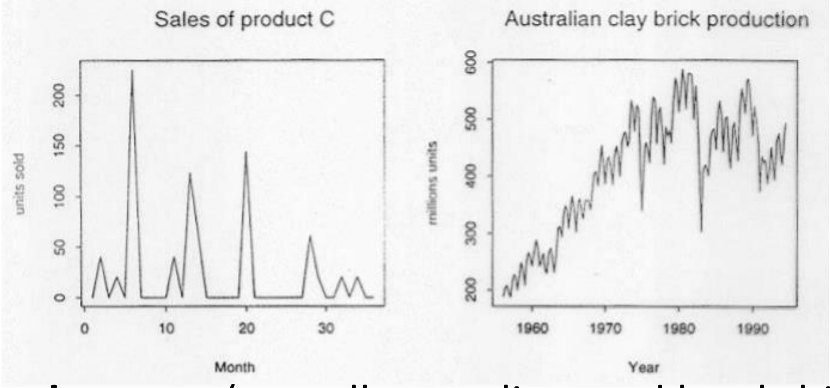




Seasonality (at year, week or day level)

Trend (Linear vs. non-linear and constant vs. changing over time)





Randomness (as well as outliers and level shifts)

Cycle (which length and intensity may differ over time)

UNIC

MOFC

Why visualize and analyze your data?

Pre-processing typically improves forecasting accuracy

Spiliotis, E., Assimakopoulos, V., Nikolopoulos, K. (2019). Forecasting with a hybrid method utilizing data smoothing, a variation of the Theta method and shrinkage of seasonal factors. International Journal of Production Economics, 209, 92-102

There are "Horses for Courses": Each forecasting method is more tailored to some types of data

Petropoulos, F., Makridakis, S., Assimakopoulos, V., & Nikolopoulos, K. (2014). 'Horses for Courses' in demand forecasting, European Journal of Operational Research, 237 (1), 152-163

You have to understand how the values of the series change over time and which factors affect these changes to select

the most appropriate forecasting approach

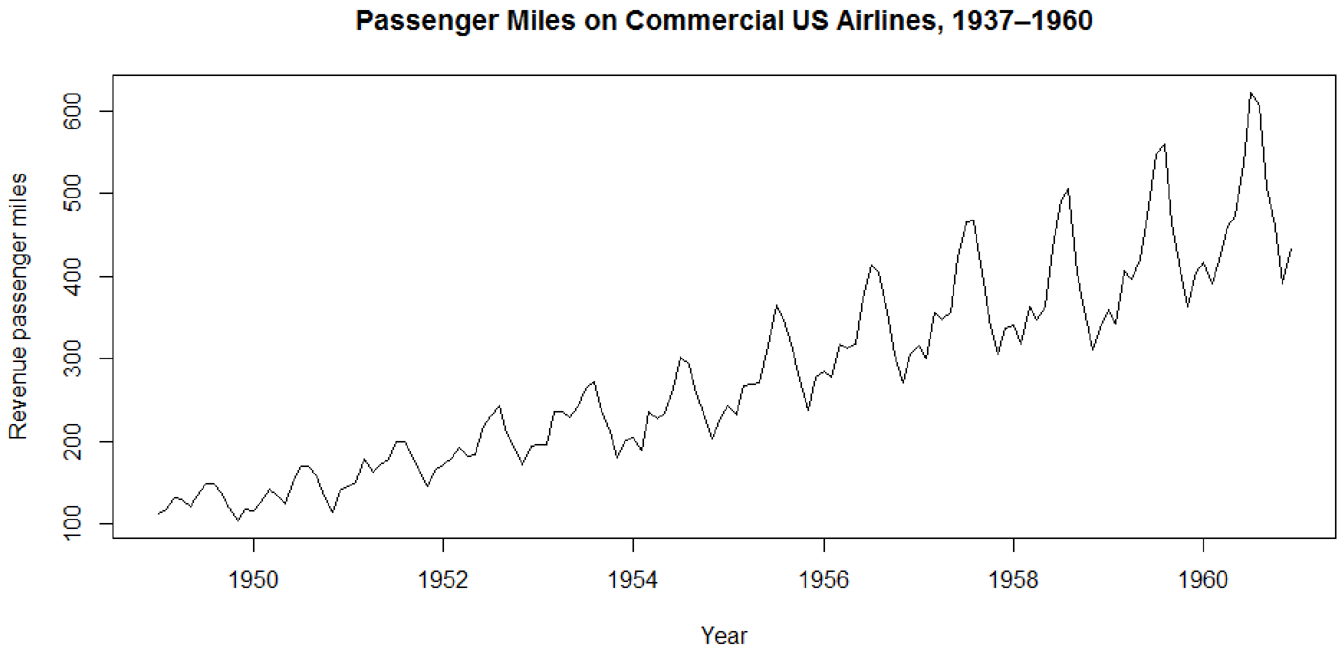
UNIC

MOFC

Visualization

```
#Plot
time_series <- AirPassengers
plot(time_series, type="l", main="Passenger Miles on Commercial US Airlines, 1937-1960",
     ylab = "Revenue passenger miles", xlab = "Year")
```

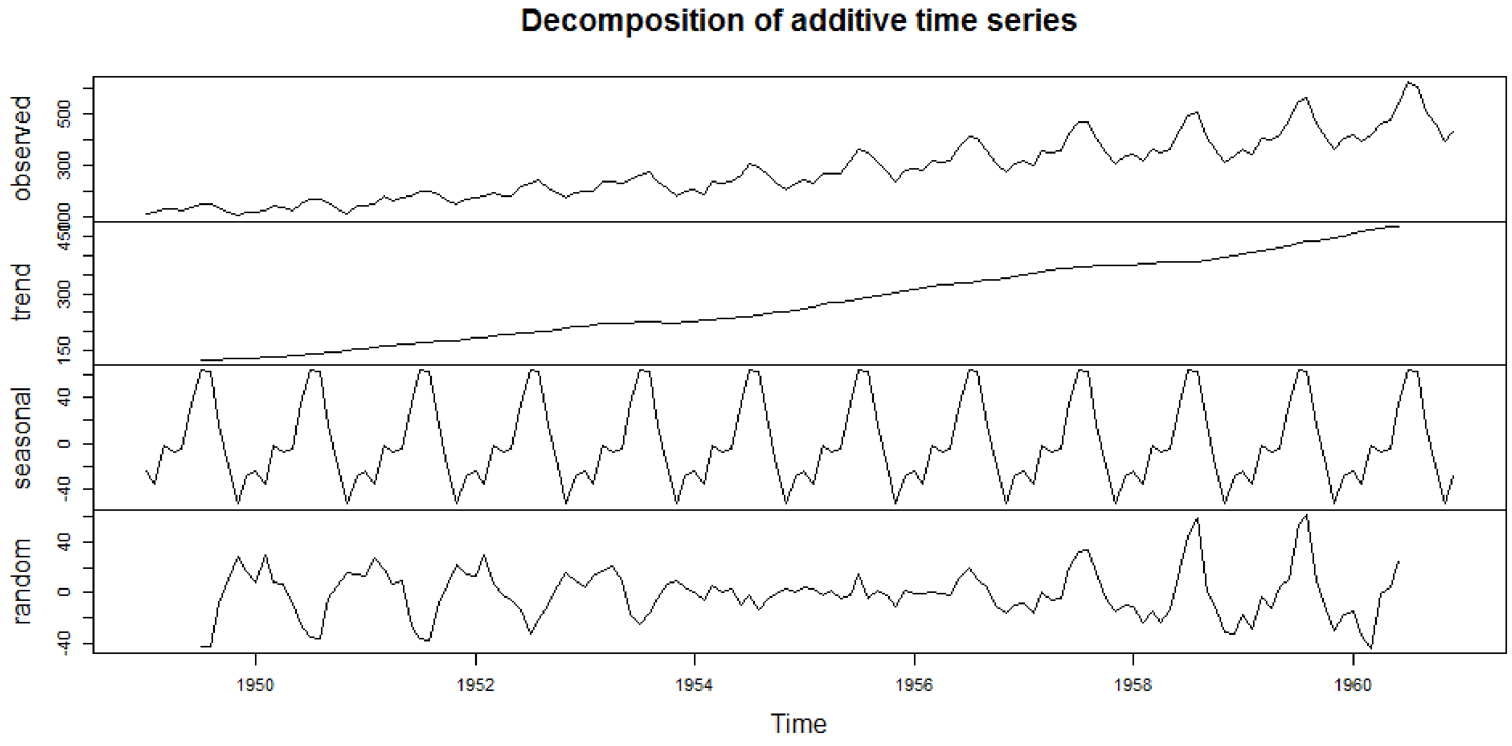
Powerful alternative to basic R plots



Decomposition

```
#Decompose
dec <- decompose(time_series, type="additive")
plot(dec)
plot(dec$seasonal[1:frequency(time_series)], type="l",
     ylab = "Index", xlab = "Period")
```

Data = Trend + Seasonal + Random

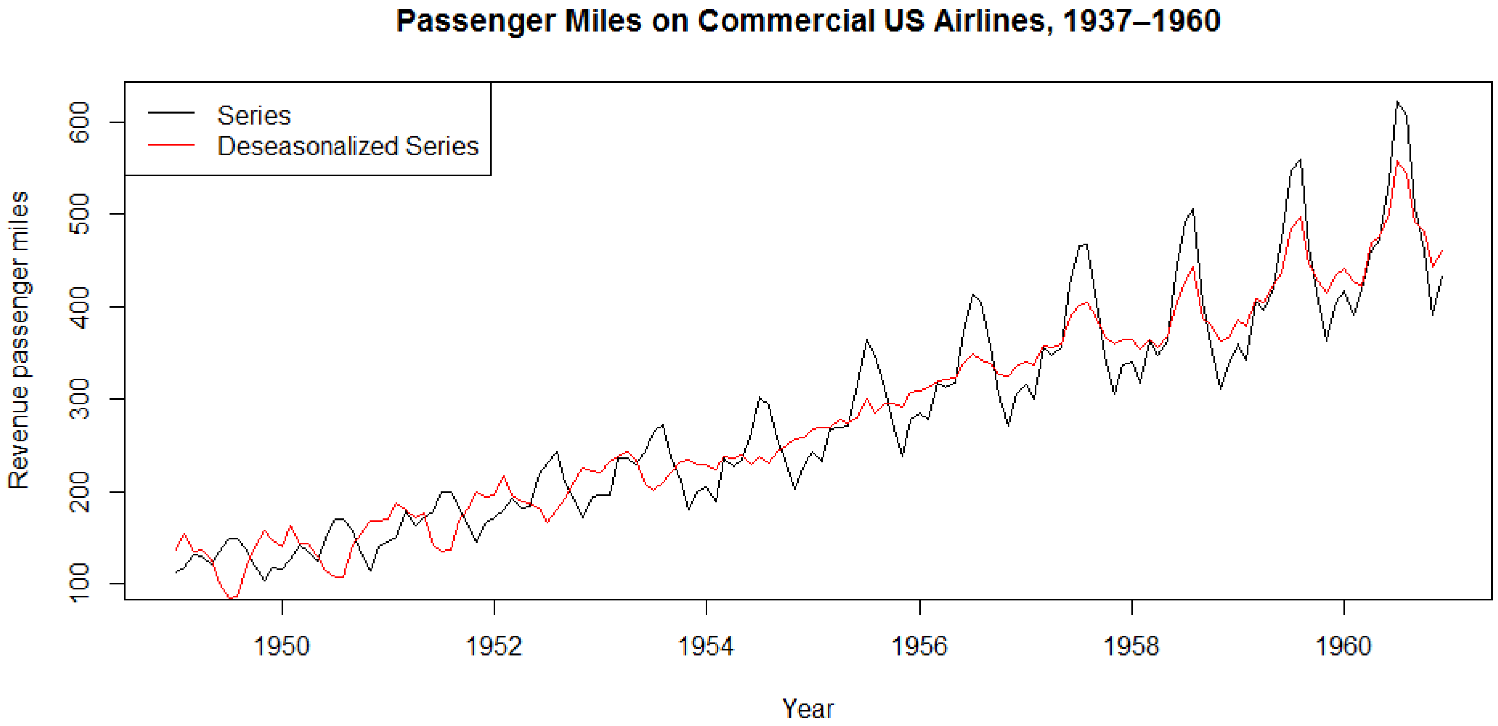


UNIC

MOFC

Additive seasonal adjustments

```
#Seasonally adjust
d_time_series <- time_series - dec$seasonal
plot(time_series, type="l", main="Passenger Miles on Commercial US Airlines, 1937-1960",
     ylab = "Revenue passenger miles", xlab = "Year")
lines(d_time_series, col="red")
legend("topleft",
       legend = c("Series", "Deseasonalized Series"),
       col = c("black", "red"), lty=1)
```



- Seasonal intensity changes over time, being subject to trend **(Heteroscedasticity)**

UNIC

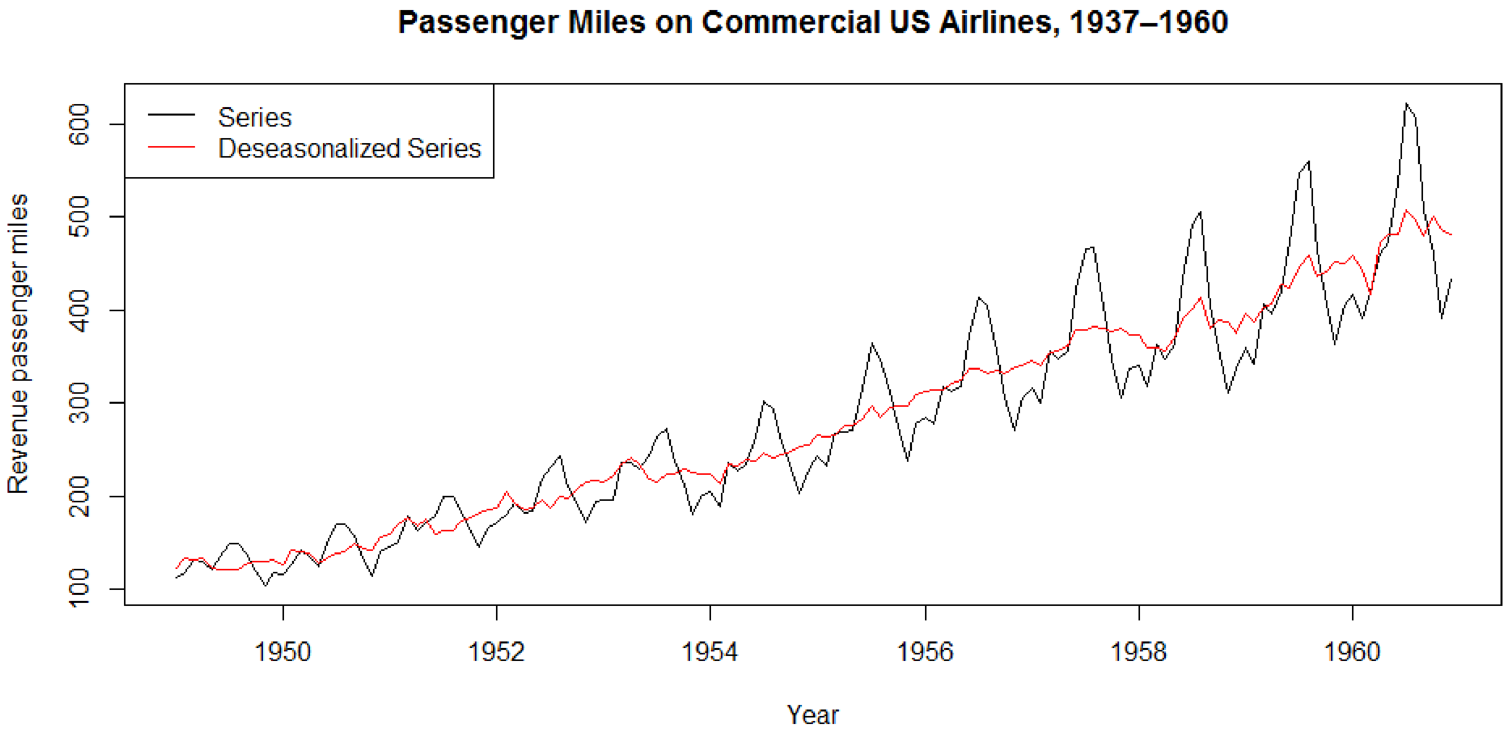
MOFC

Multiplicative seasonal adjustments

```
#Seasonally adjust
dec <- decompose(time_series, type="multiplicative")
d_time_series <- time_series / dec$seasonal
plot(time_series, type="l", main="Passenger Miles on Commercial US Airlines, 1937-1960",
     ylab = "Revenue passenger miles", xlab = "Year")
lines(d_time_series, col="red")
legend("topleft",
       legend = c("Series", "Deseasonalized Series"),
       col = c("black", "red"), lty=1)
```

Data = Trend * Seasonal * Random

- The series is characterized by **multiplicative** seasonality



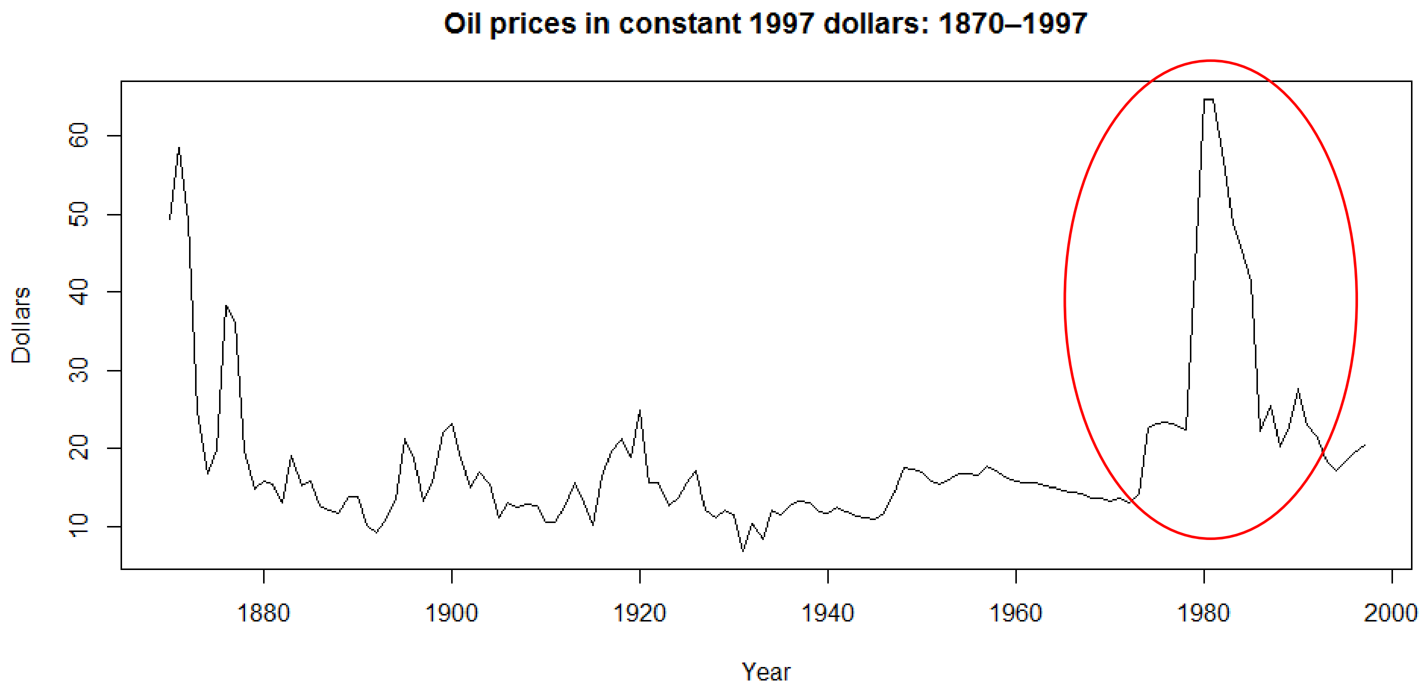
UNIC

MOFC

Distribution of data (1/2)

```
library(fpp)
plot(oilprice, type="l", ylab="Dollars", xlab = "Year",
     main="Oil prices in constant 1997 dollars: 1870–1997")
```

- The 1980s oil glut was a serious surplus of crude oil caused by falling demand following the 1970s energy crisis

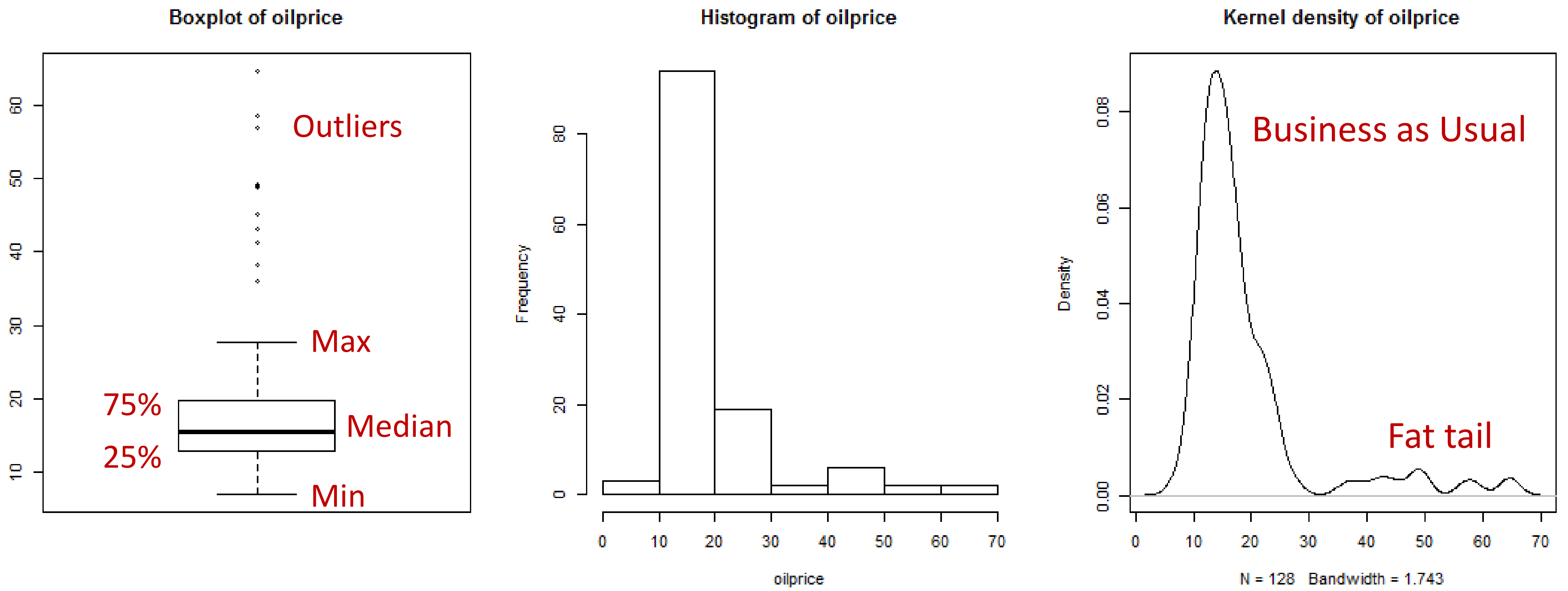


UNIC

MOFC

Distribution of data (2/2)

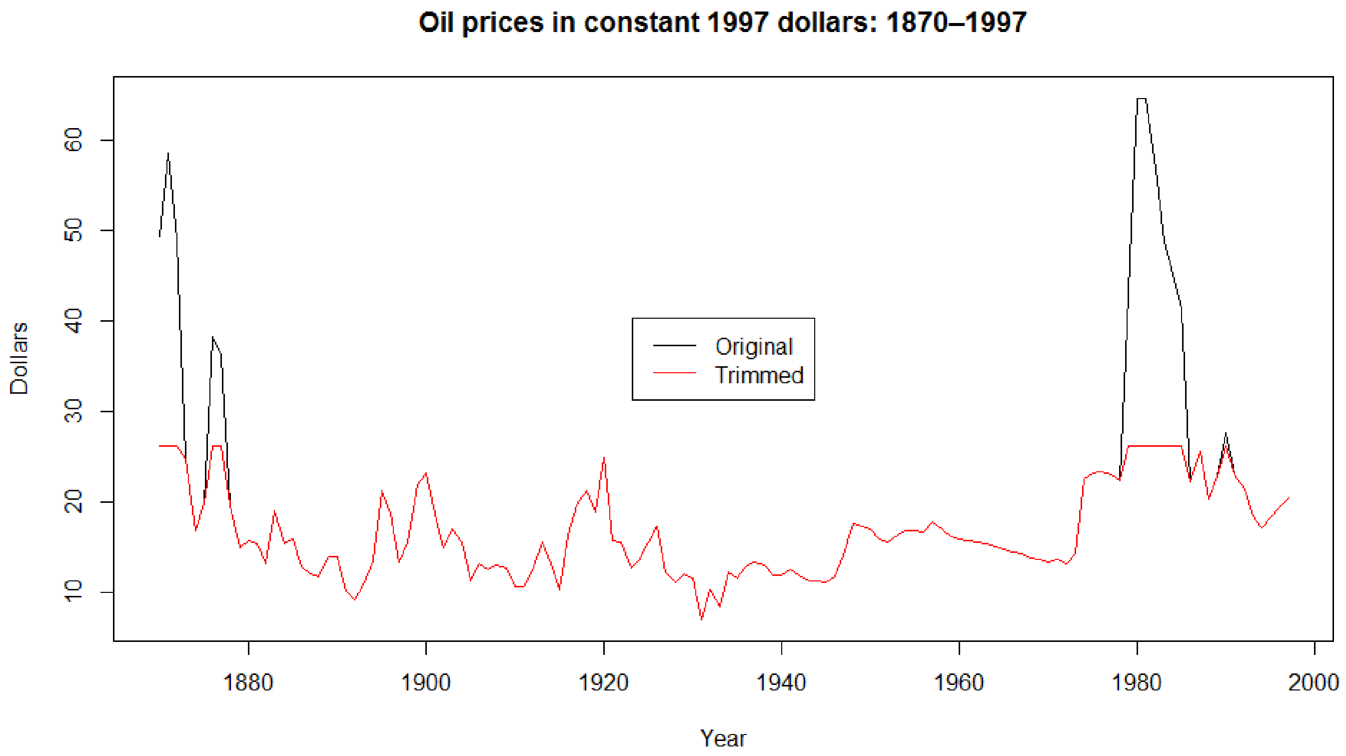
```
library(fpp)
plot(oilprice, type="l", ylab="Dollars", xlab = "Year",
     main="Oil prices in constant 1997 dollars: 1870-1997")

par(mfrow=c(1,3))
boxplot(oilprice, main="Boxplot of oilprice")
hist(oilprice)
plot(density(oilprice), main="Kernel density of oilprice")
```



Deal with outliers – By trimming

```
#Remove outliers
limit <- quantile(oilprice, 0.90)
n_oilprice <- oilprice
n_oilprice[n_oilprice>limit] <- limit
plot(oilprice, type="l", ylab="Dollars", xlab = "Year",
     main="Oil prices in constant 1997 dollars: 1870-1997")
lines(n_oilprice, col="red")
legend("center",
       legend = c("Original", "Trimmed"),
       col = c("black", "red"), lty=1)
```

- The limit is **arbitrarily** set to the top 10% of the observed values
- The same can be down for low prices

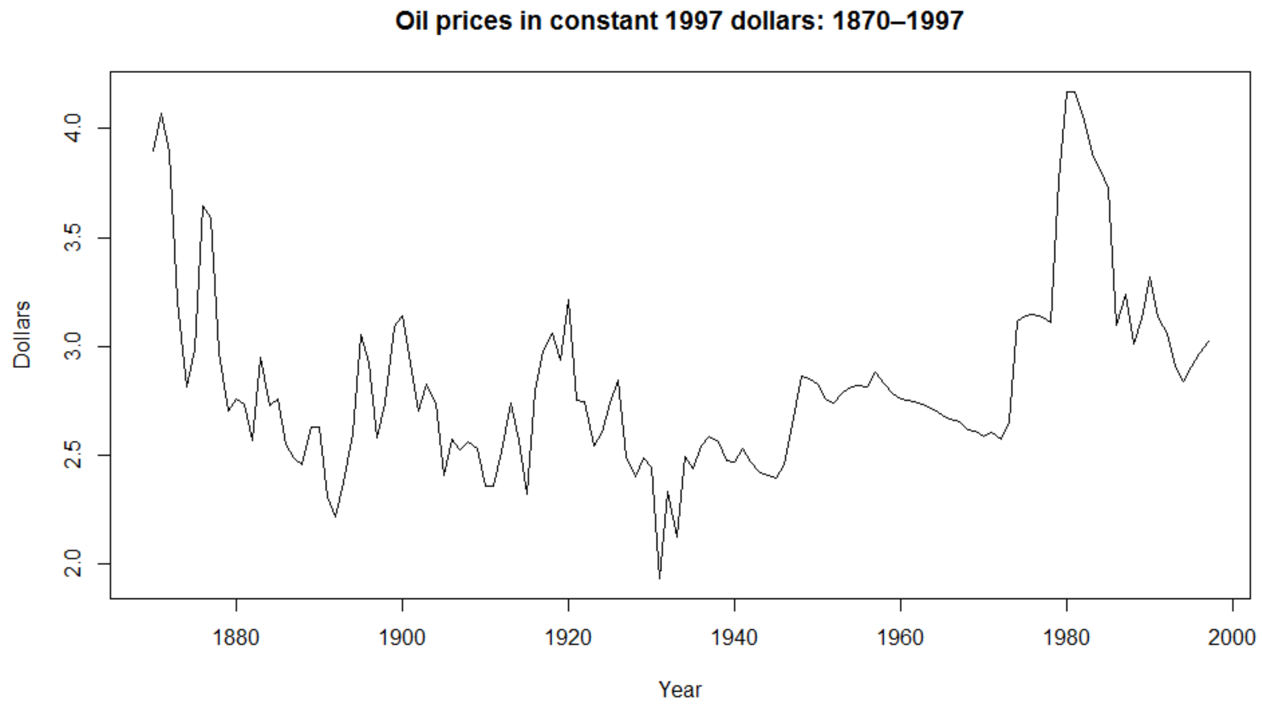


UNIC

MOFC

Deal with outliers – By reducing variance

```
plot(log(oilprice), type="l", ylab="Dollars", xlab = "Year",
     main="Oil prices in constant 1997 dollars: 1870-1997")

sd(oilprice)*100/mean(oilprice)
sd(log(oilprice))*100/mean(log(oilprice))
```



- Although the outliers are still visible, their extent has been significantly reduced
- **Coefficient of Variation (CV)**
  - ✓ Before: 58.65%
  - ✓ After: 15.05%

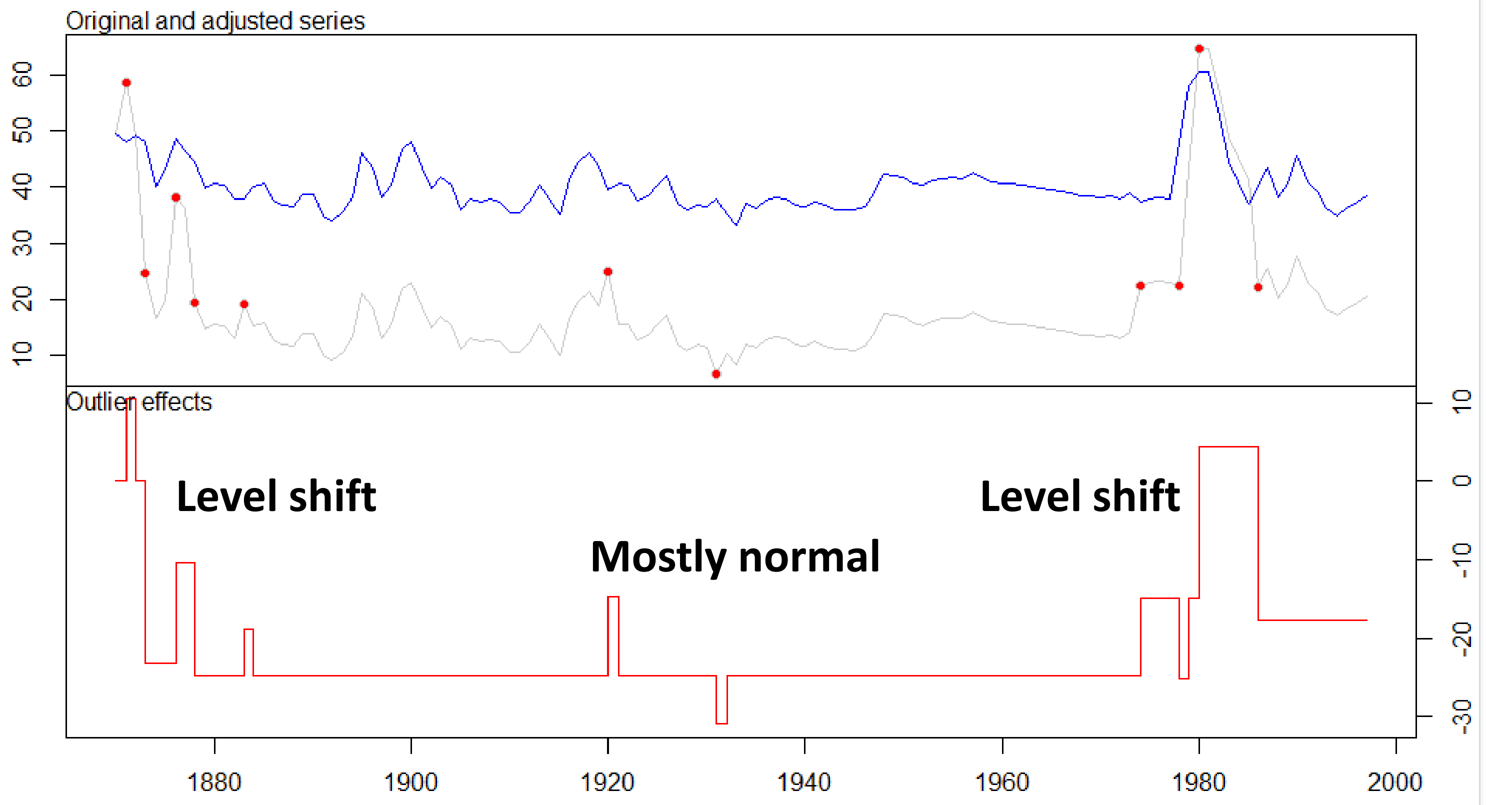
UNIC

MOFC

Deal with outliers – By fitting an ARIMA model

```
library(tsoutliers)
product.outlier<-tso(oilprice,types=c("AO","LS"), maxit.iloop = 6)
plot(product.outlier)
```

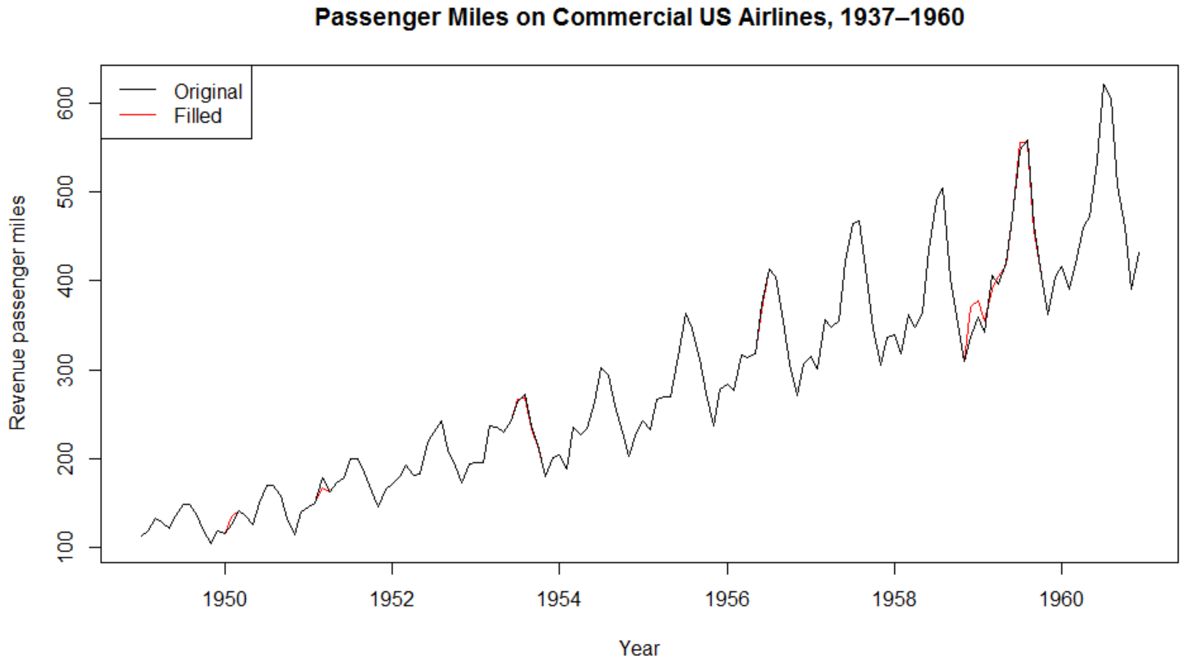
```
Outliers:
   type ind time coefhat   tstat
1    AO   2 1871  10.430   5.534
2    LS   4 1873 -23.272  -8.297
3    LS   7 1876  12.928   5.218
4    LS   9 1878 -14.492  -5.598
5    AO  14 1883   5.931   5.099
6    AO  51 1920  10.110   7.190
7    AO  62 1931  -6.220  -4.451
8    LS 105 1974   9.971   5.112
9    AO 109 1978 -10.458  -7.448
10   LS 111 1980  19.207   8.069
11   LS 117 1986 -22.165 -10.652
```



Deal with missing values

```
#Fill missing values
missing <- AirPassengers
missing[c(14,27,55,56,57,58,90,120,121,122,
          123,124,125,127,128,129)] <- NA

for (i in (frequency(missing)+1):(length(missing)-12)){
  if (is.na(missing[i])==TRUE){
    missing[i] <- mean(c(missing[i-frequency(missing)], missing[i+frequency(missing)]))
  }
}

plot(missing, type="l", main="Passenger Miles on Commercial US Airlines, 1937-1960",
     ylab = "Revenue passenger miles", xlab = "Year", col="red")
lines(AirPassengers, col="black")
legend("topleft",
       legend = c("Original", "Filled"),
       col = c("black", "red"), lty=1)
```

- **Non-seasonal:** Average of the previous and the following observations
- **Seasonal & non-trended**: Average of all the observations of the same period
- **Seasonal & trended**: Average of the previous and following observations of the same period



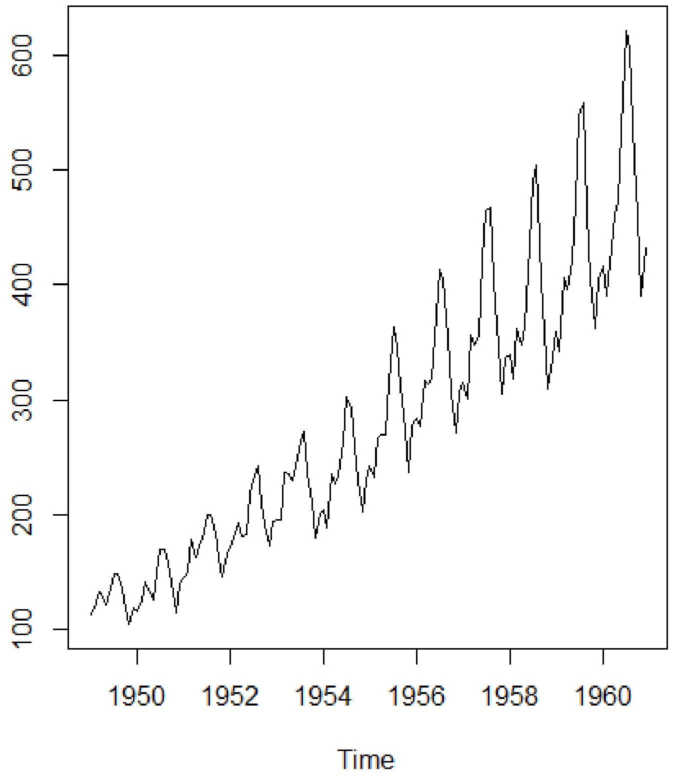
UNIC

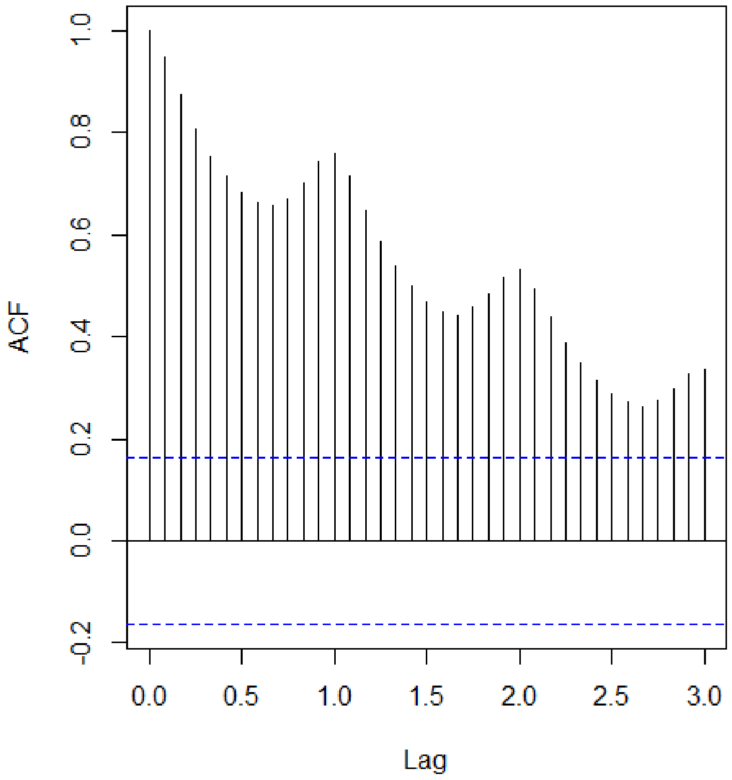
MOFC

Autocorrelation (1/4)

```
#Correlation between observations
par(mfrow=c(1,2))

plot(time_series)
acf(time_series,lag.max=36)
```

- Correlation decreases through time -> **Trend**
- Correlation oscillates every 12 periods -> **Seasonality**





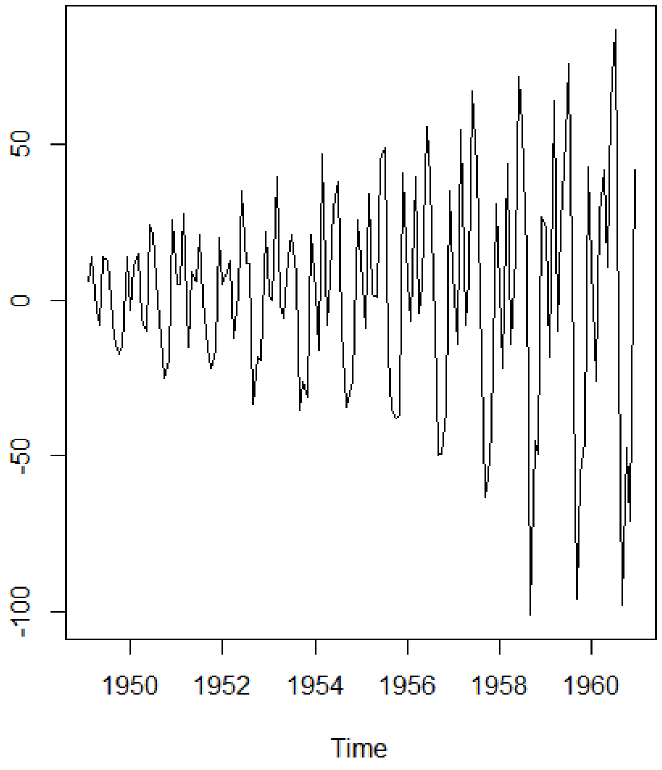
UNIC

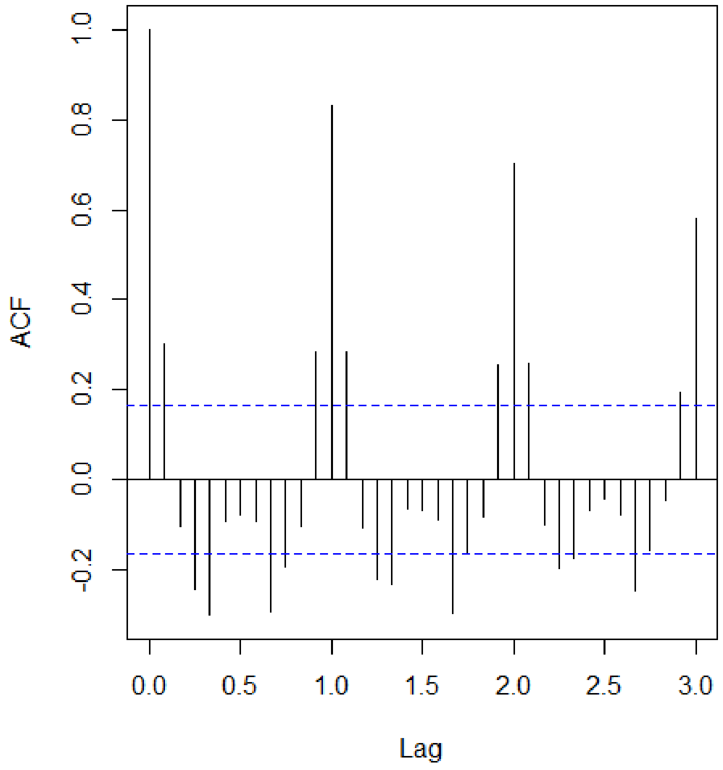
MOFC

Autocorrelation (2/4)

```
plot(diff(time_series,1))
acf(diff(time_series,1),lag.max=36)
```

- **First differences:** Trend has been removed– Seasonality is still strong





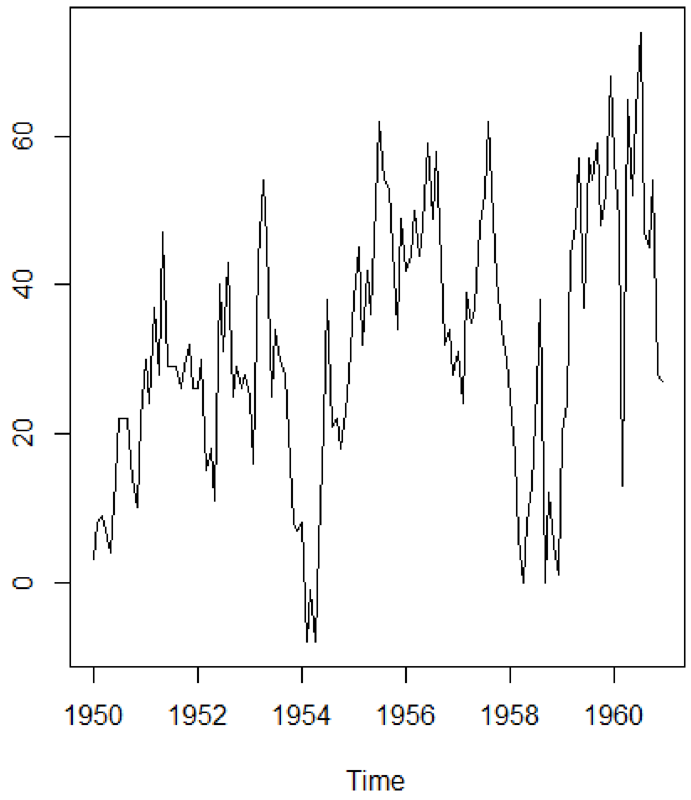
UNIC

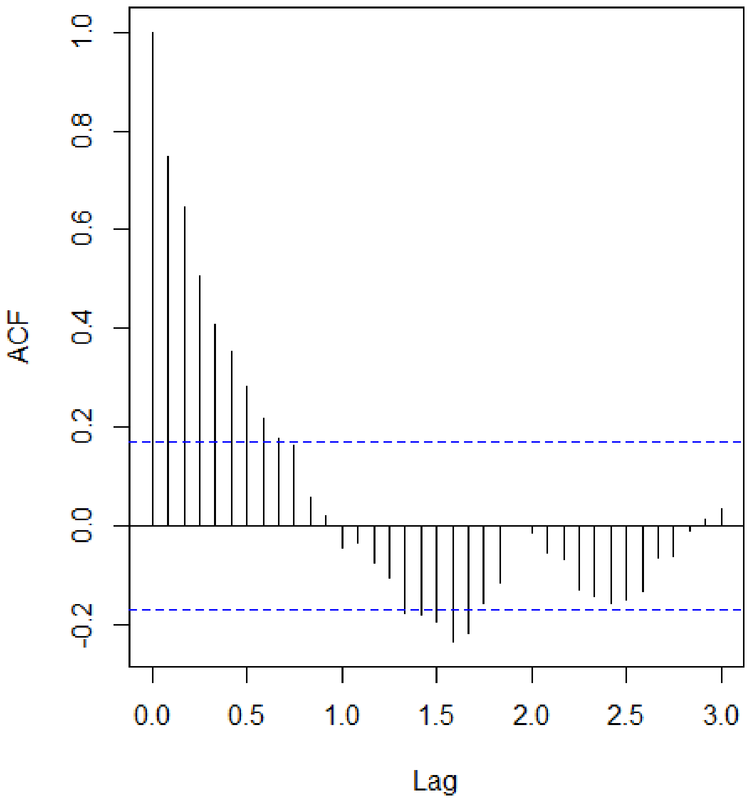
MOFC

Autocorrelation (3/4)

```
plot(diff(time_series,12))
acf(diff(time_series,12),lag.max=36)
```

- **Seasonal differences:** Seasonality has been removed– Trend is still observable





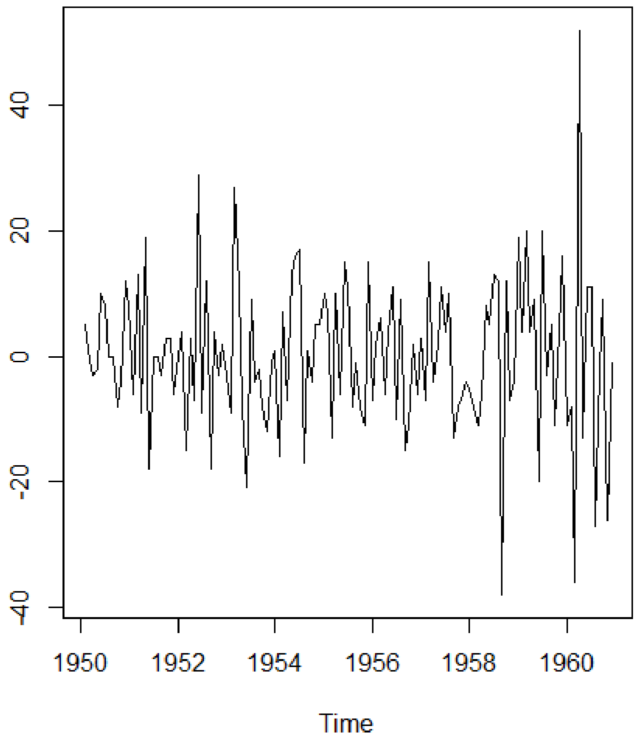
UNIC

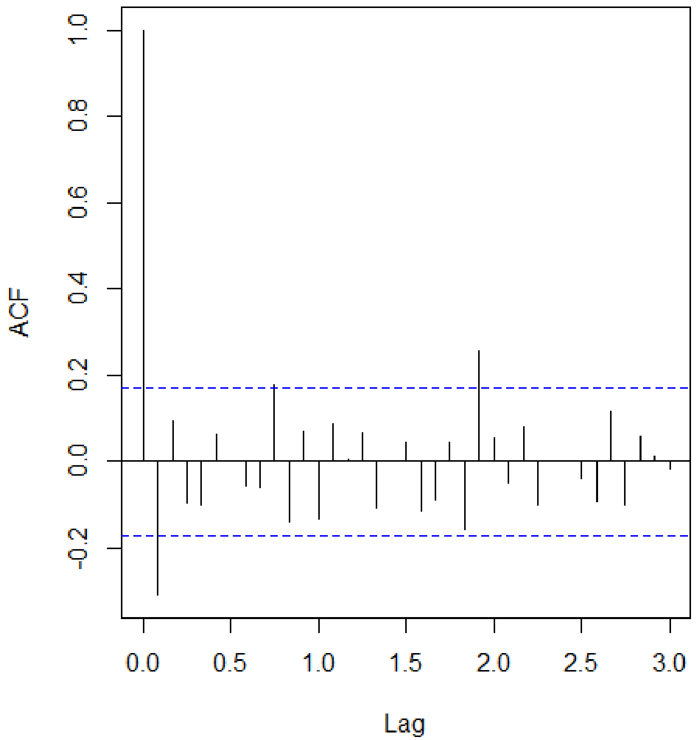
MOFC

Autocorrelation (4/4)

```
plot(diff(diff(time_series,12),1))
acf(diff(diff(time_series,12),1),lag.max=36)
```

- **First and Seasonal differences:** We get a stationary series (mean and deviation constant through time)





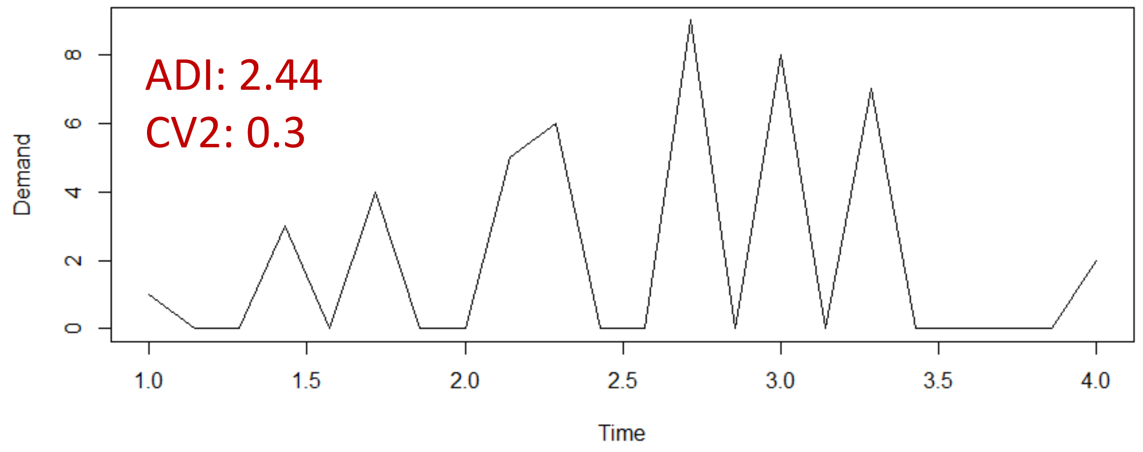
- Differentiation is another way for **decomposing** a series
- Useful for making time series data ready-to-be used by **ML forecasting methods** (*typically assume stationarity*)

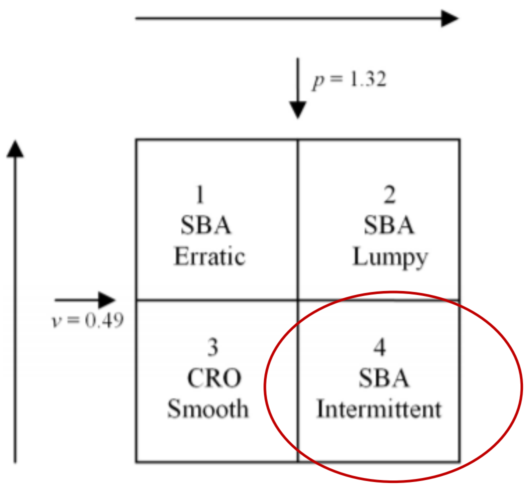
UNIC

MOFC

Intermittent demand data

```
#Intermittent demand
ts_int <- ts(c(1,0,0,3,0,4,0,0,5,6,0,
               0,9,0,8,0,7,0,0,0,0,2), frequency = 7)
plot(ts_int, ylab="Demand")

demand <- ts_int[ts_int!=0]
interval <- c(1) ; counter <- 1
for (i in 2:length(ts_int)) {
  if (ts_int[i]==0){
    counter <- counter + 1
  }else{
    interval <- c(interval, counter)
    counter <- 1
  }
}
stats <- data.frame(demand, interval)

mean(stats$interval) #ADI
(sd(stats$demand)/mean(stats$demand))^2 #CV2
```





UNIC

MOFC

Forecasting and Uncertainty

Week 2: Data for Forecasting

UNIVERSITY of NICOSIA

Forecasting Methods

Data Generation Process is assumed a priori

Statistical	Machine Learning
• Naïve • Moving Averages • Exponential Smoothing • ARIMA	• Neural Networks • Regression Trees • Support Vector Regression • K-Nearest Neighbor regression • Gaussian Processes

We assume nothing and data relations are being learned

UNIC

Makridakis S., Spiliotis E. & Assimakopoulos V. (2018). Statistical and machine learning forecasting methods: concerns and ways forward. PLoS One, 13 (3), 1-26

MOFC

Statistical Forecasting Methods

- **Naive**
  - **Naïve 1**: The forecast is the same to the last observation of the series
  - **Naïve 2**: Like Naïve 1, but using seasonally adjusted data
  - **Seasonal Naïve**: The forecast is the same to the last observation of the series of the same period
  - **Drifted Naïve**: Like Naïve 1, but trend (*average of first differences*) is added at every step

- **Moving Averages**: The forecast is the average of the last *k* observations of the series

- **Exponential Smoothing**: Like Moving Averages, but this time all observations are used with exponential weights (*more emphasis is given to the last observations*). Can be seasonal, trended or both.

- **Theta**: Similar to exponential smoothing, but with drift (*linear regression of data over* time)

- **ARIMA**: Forecasts are given as functions to the last *p* observations of the series and the last *q* errors produced by the model. Differences are used to achieve stationarity.

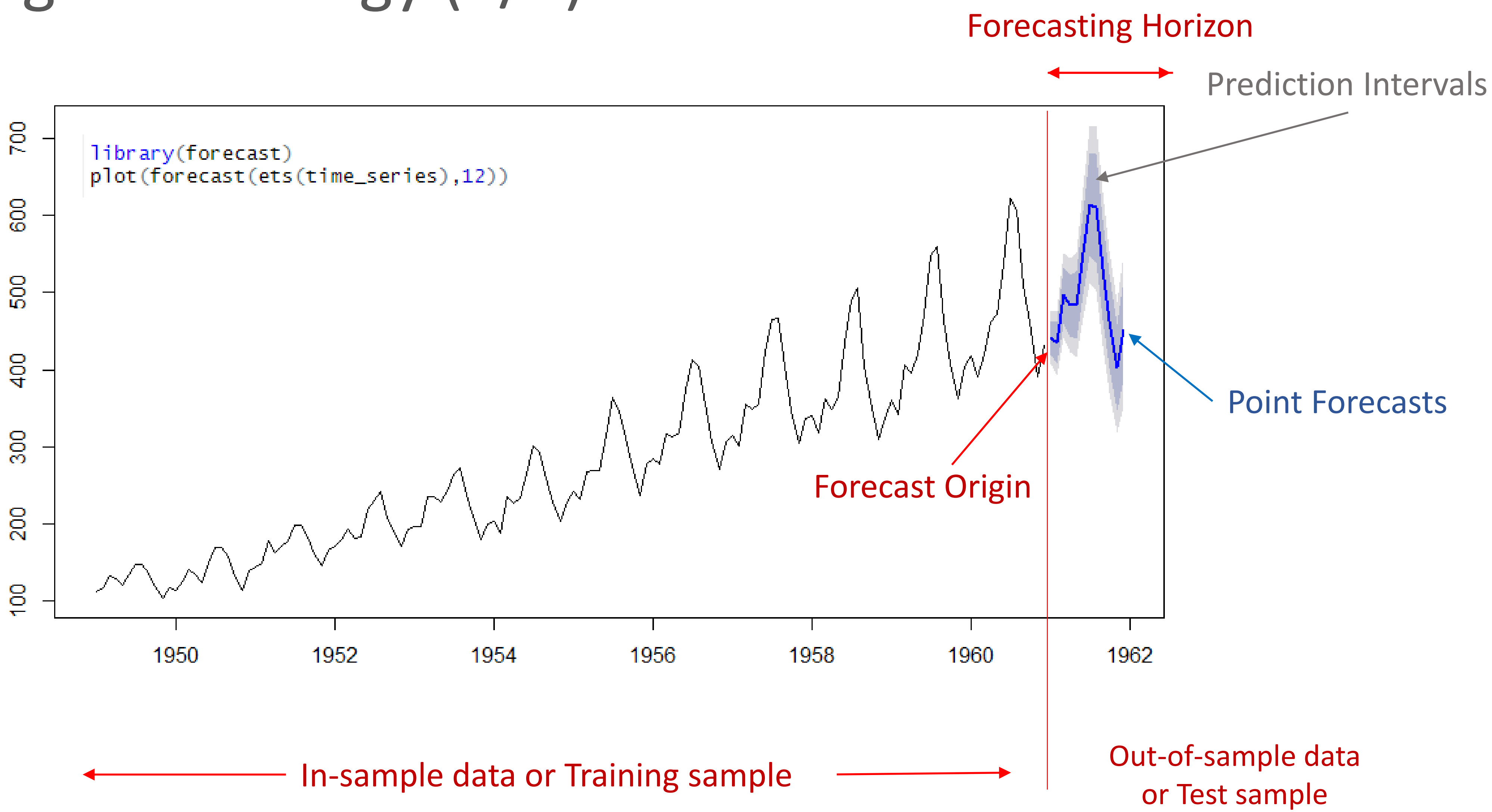
UNIC

MOFC

Machine Learning Forecasting Methods

- **Neural Networks**: Like AR models, forecasts are given as functions to the last *p* observations of the series. However, the relations can be both linear and non-linear
- **Regression Trees**: Rules are determined and sequentially used to define the best forecast based on the values of the inputs provided for training
- **Support Vector Regression**: Divides the space of solutions so that the margin between two classes is maximized and the total error is minimized
- **K-Nearest Neighbor regression**: The forecast is equal to the average of the *k* observations used for training that look more similar to the one provided for predicting
- **Gaussian Processes**: Forecasts are associated with one or more normally distributed random variables which form a multivariate normal distribution, emerging by combining the individual distributions of the independent ones. Non parametric regression is then used for deriving the forecasts.

Forecasting terminology (1/2)



UNIC

MOFC

Forecasting terminology (2/2)

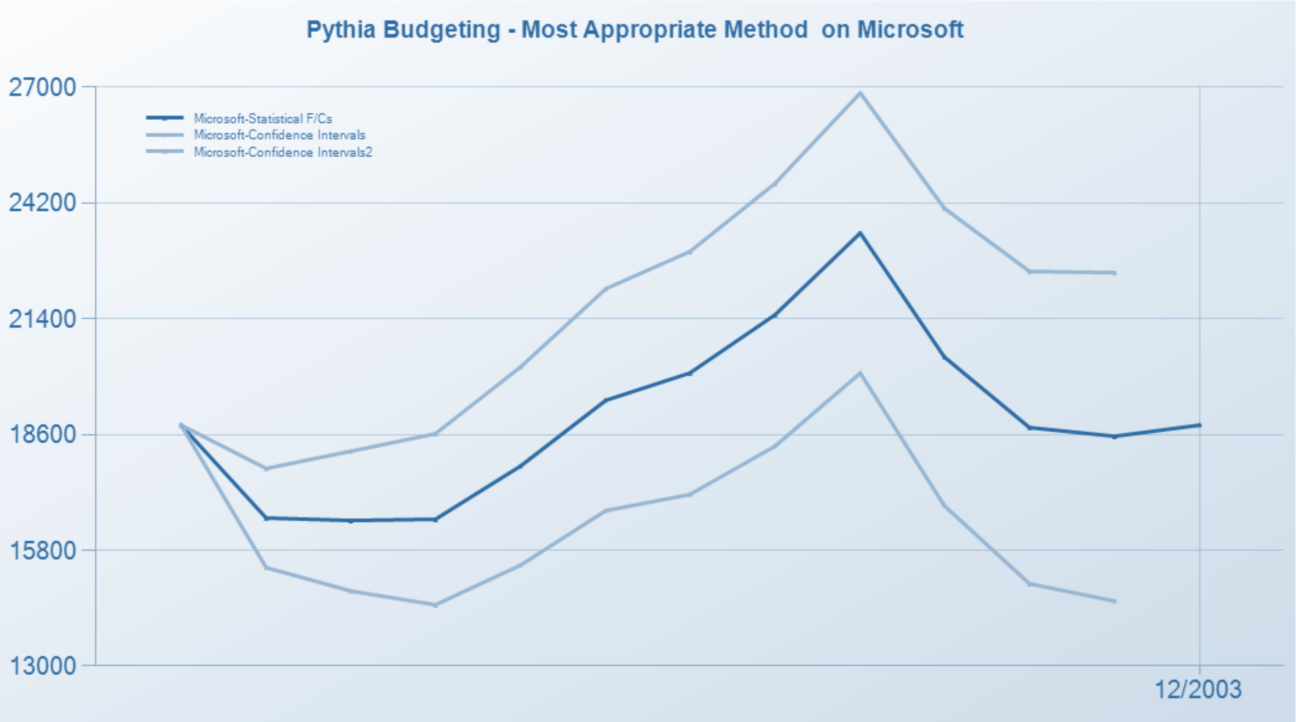
- Point Forecasts denote what is “**most likely to happen**”
- Assuming that future data are normally distributed, point forecasts should approximate the **mean** of the distribution
- A **prediction interval** of a% indicates that the a% of the future data should lie within the upper and lower specified bounds
- Accordingly, a **probabilistic forecast** provided for quantile u, indicates that the u*100% of the future data should be lower than the specified bound
- In practice, future data **are not distributed normally**, meaning that prediction intervals are hard to specify and, accordingly, the uncertainty present is difficult to capture

Probabilistic forecasts (1/2)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - F_i)^2}$$

$$F_i = F_i \pm t \cdot RMSE \cdot \sqrt{i-n}$$

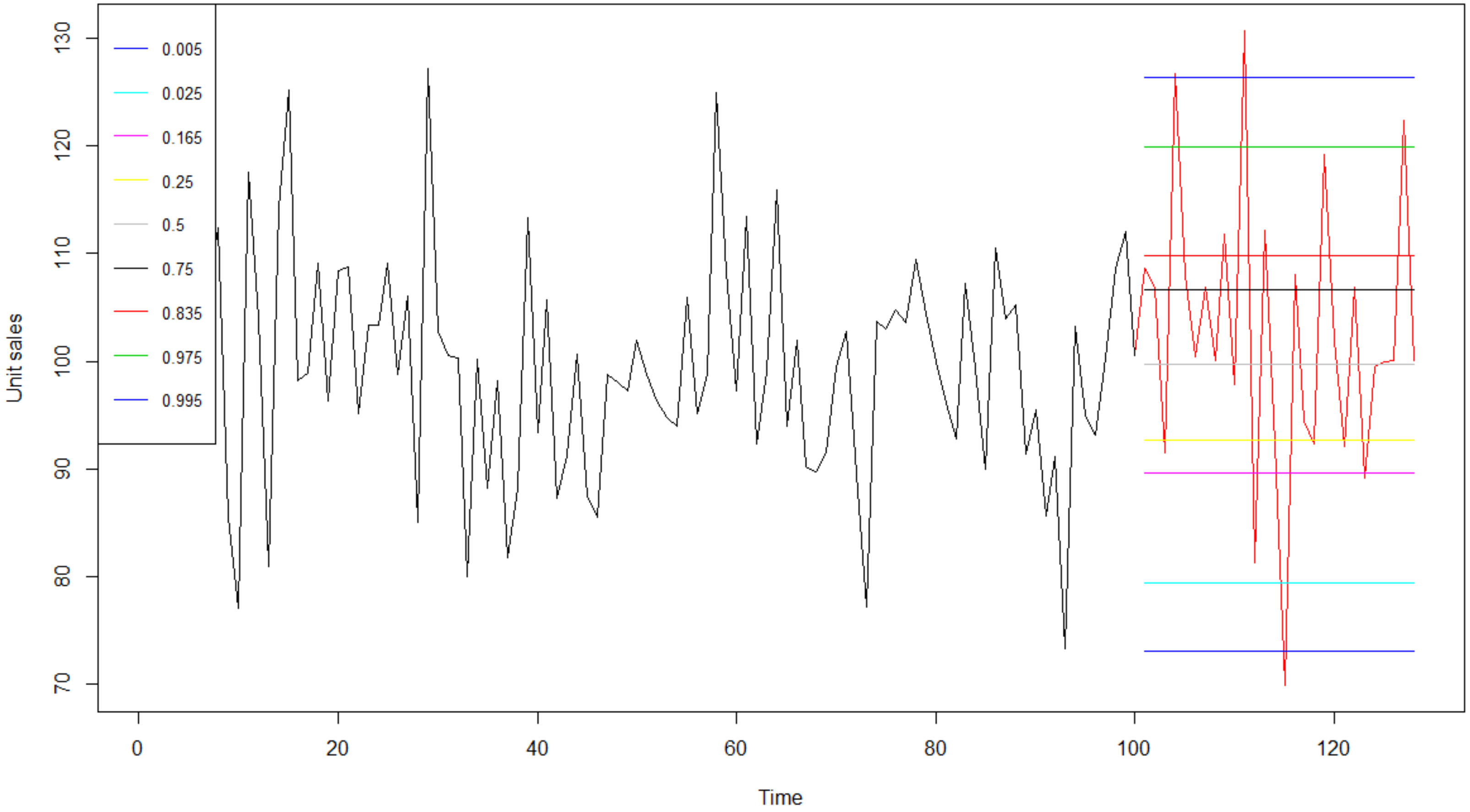
Confidence	t
99%	2.580
98%	2.330
95%	1.960
90%	1.645
80%	1.280



UNIC

MOFC

Probabilistic forecasts (2/2)



UNIC

MOFC

Things to keep in mind…

- **Longer forecasting horizons** mean **more uncertainty** about the future and, therefore, **higher forecast errors**
  - ✓ *Data frequency is also important*: The uncertainty is larger for one-step-ahead yearly forecasts than three-step-ahead monthly forecasts

- **Longer training samples typically** mean more information about the series and, therefore, **better forecasts**

- In-sample forecasting accuracy is not always related to out-sample one
  - ✓ Over-fitting
  - ✓ One-step-ahead forecasts for training but Multi-step-ahead for forecasting
  - ✓ **Data uncertainty** (*data patterns may change*)
  - ✓ **Model uncertainty** (*maybe you haven't chosen the best model*)
  - ✓ **Parameter uncertainty** (*even if your model is right, its parameters may not be appropriate*)

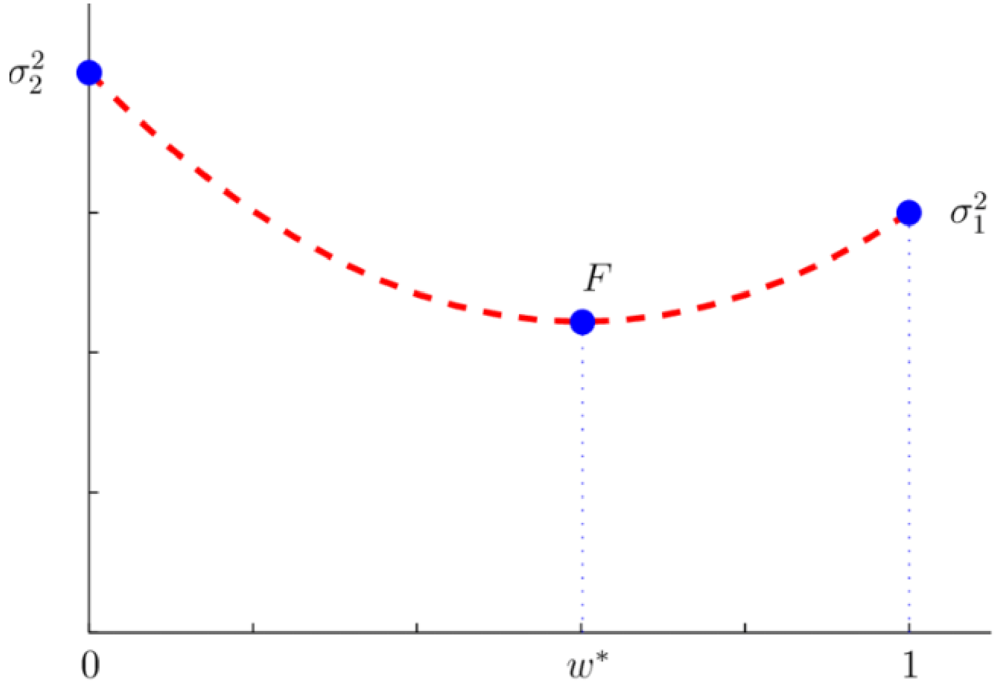
Paradox: We know that trusting the past for predicting the future is wrong but we keep doing that since we do not have anything else to work with

UNIC

MOFC

Mitigate Uncertainty (1/3)

Combine different forecasting methods: Many – Good – Diverse. All models are wrong, so why bet just on a single one? Individual errors are typically canceled by averaging.

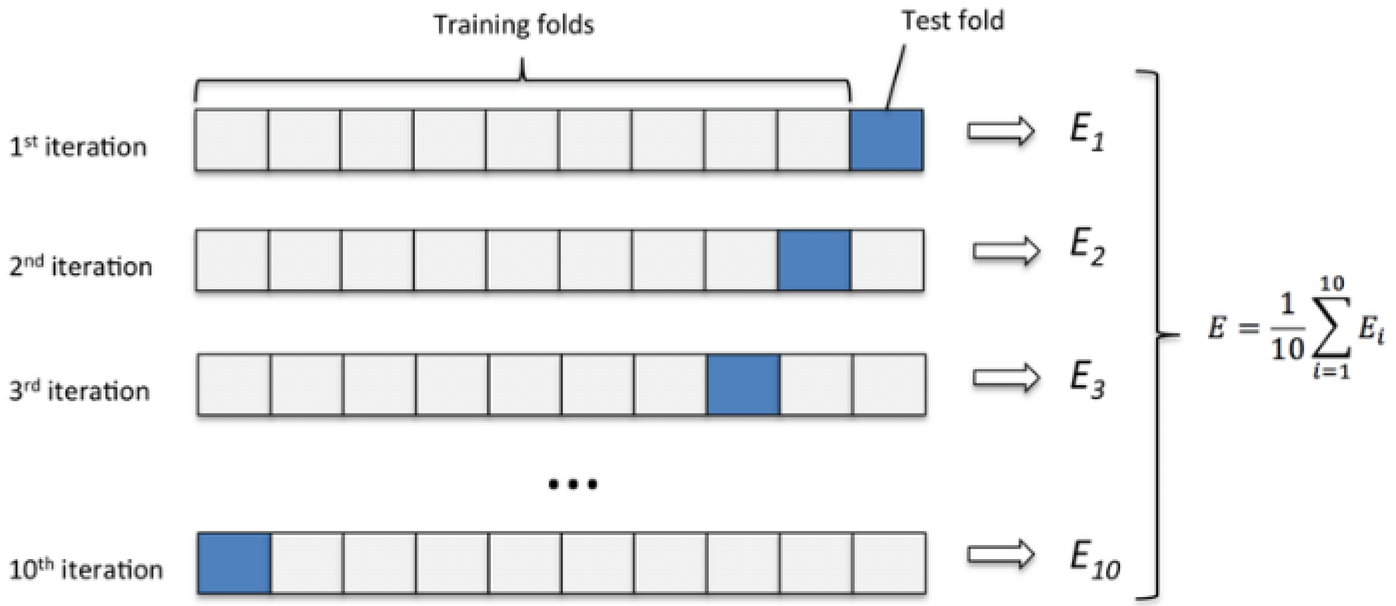


Fildes, R. & Petropoulos, F. (2015). Simple versus complex selection rules for forecasting many time series. Journal of Business Research, 68 (8), 1692-1701

Mitigate Uncertainty (2/3)

Use cross-validation: Test the performance of the various alternatives over multiple parts of the training sample (also see rolling origin evaluation)

Can be used both for model selection, weighted combinations of models, and parameter estimation



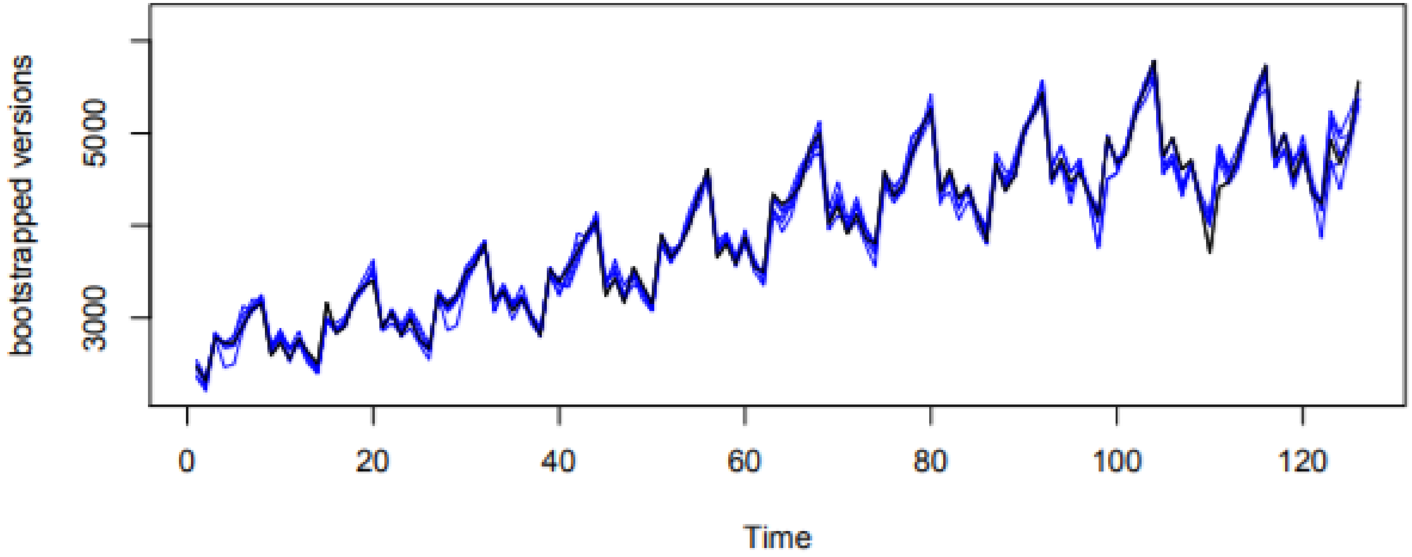
UNIC

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. International Journal of Forecasting, 16 (4), 437-450

MOFC

Mitigate Uncertainty (3/3)

Bagging: Test the performance of the various alternatives over multiple versions of the series being predicted
Can be used both for model selection and parameter estimation



Petropoulos, F., Hyndman, R. J. & Bergmeir, C. (2018). Exploring the sources of uncertainty: Why does bagging for time series forecasting work?. European Journal of Operational Research, 268 (2), 545-554

UNIVERSITY *of* NICOSIA