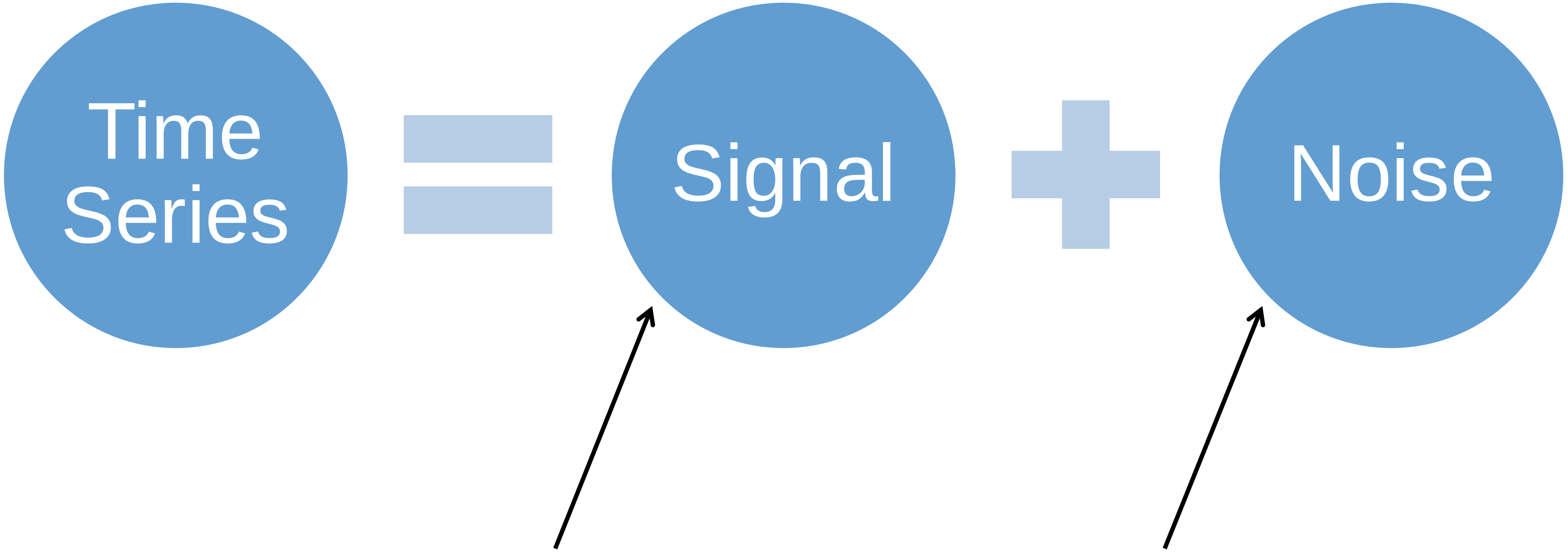
# SEASONALITY MODELS

Dr. Aric LaBarr

Institute for Advanced Analytics

# QUICK REVIEW

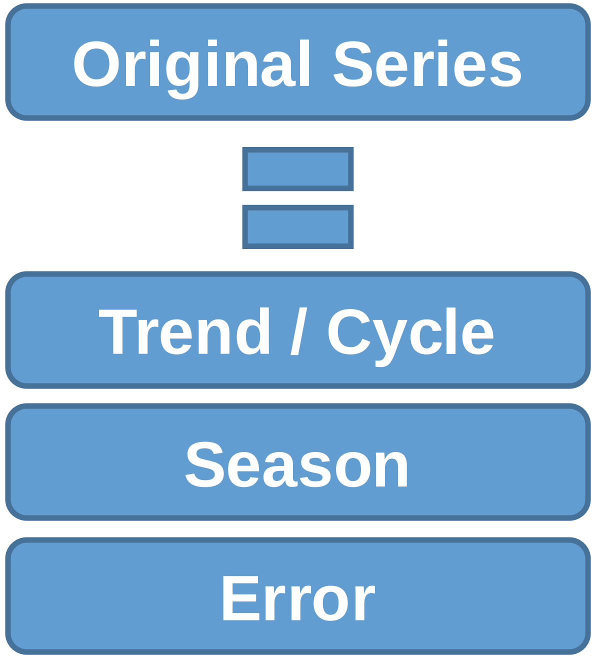
# Time Series Data

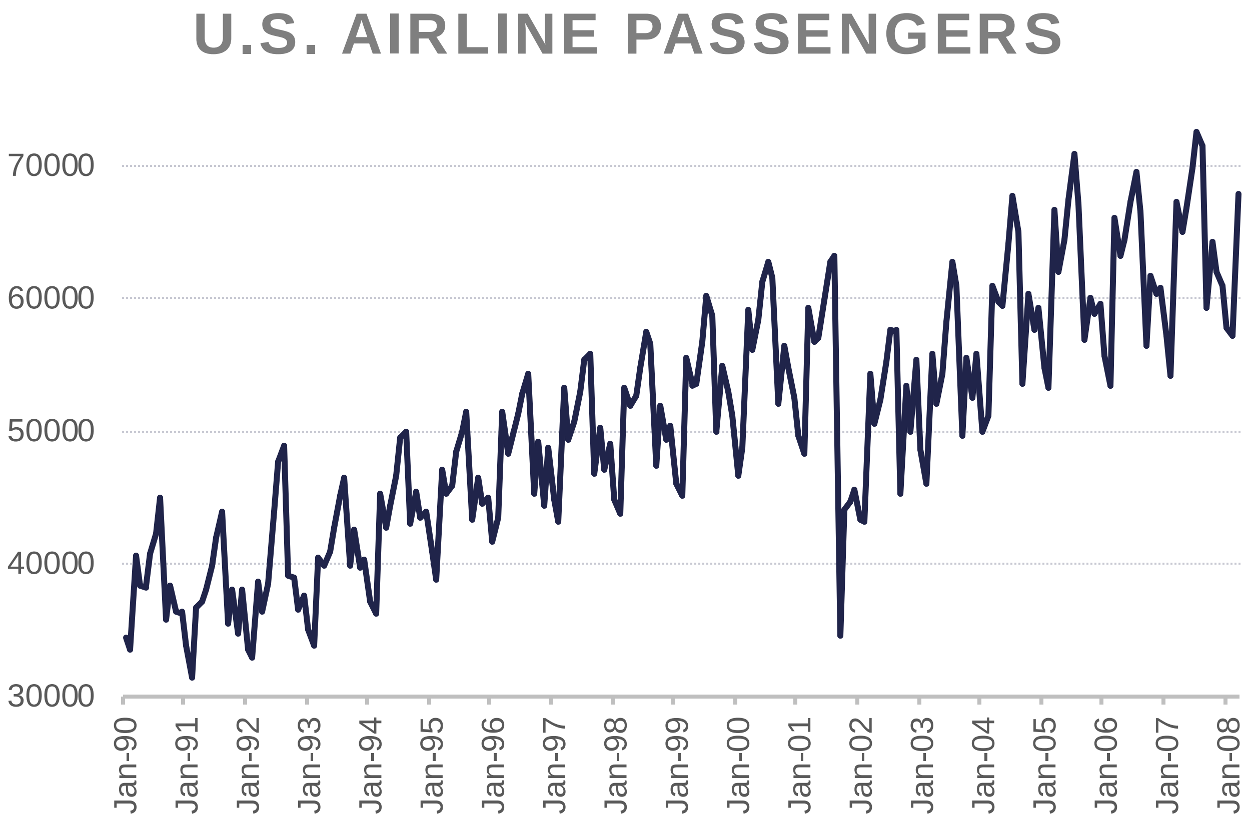



Forecasts extrapolate signal portion of model

Confidence intervals account for uncertainty

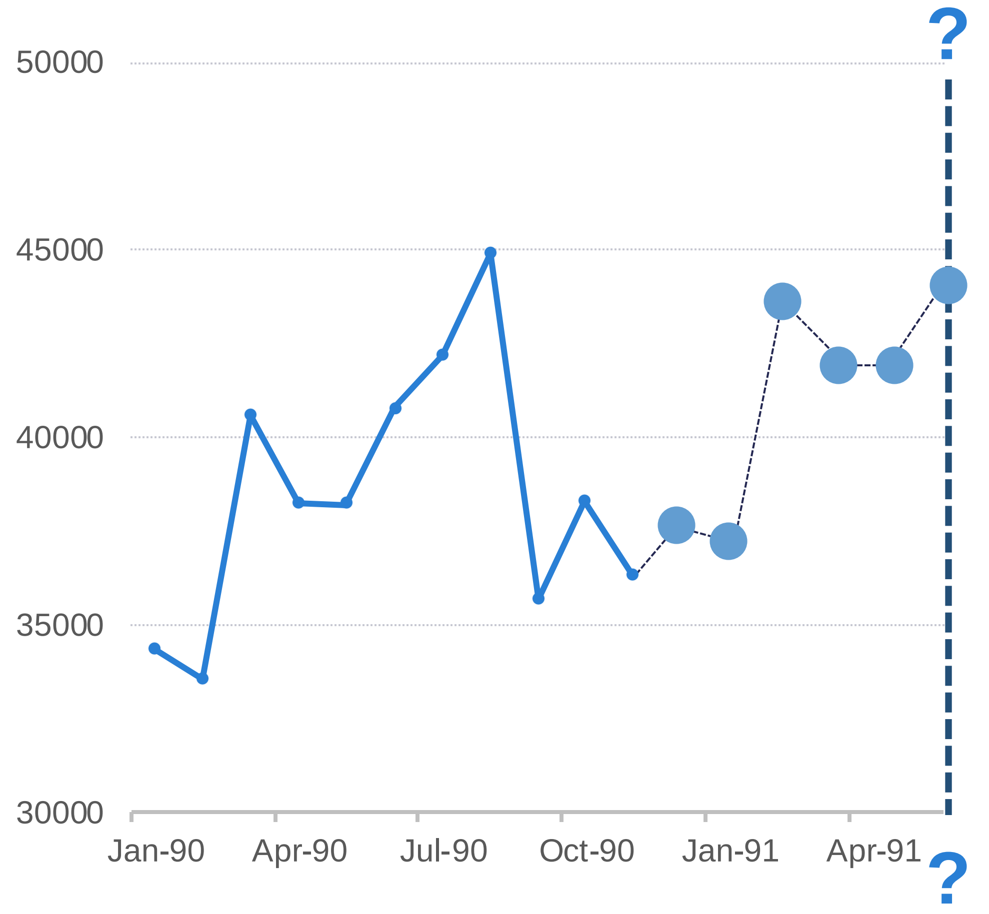
# Time Series Data



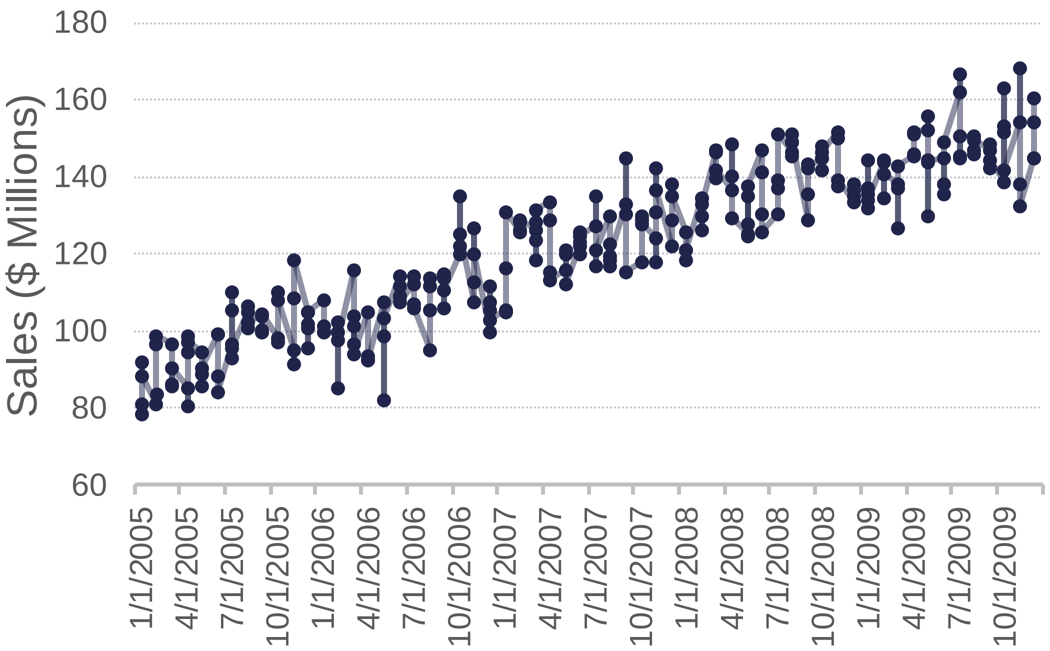
# Exponential Smoothing Models

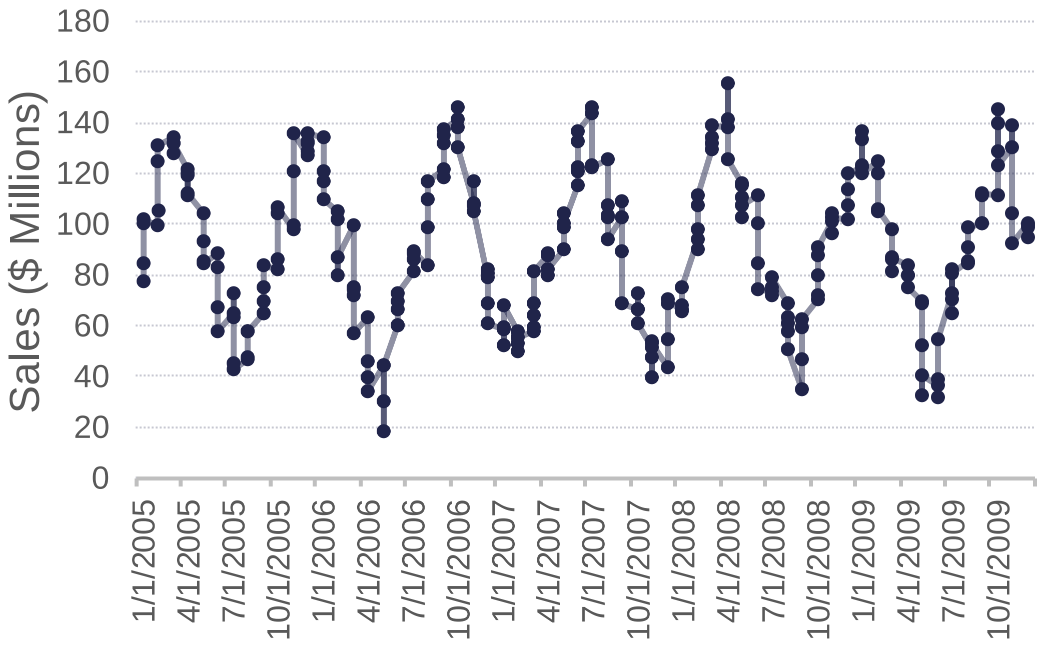
| | |
|---|---|
| **Original Series** | $\hat{Y}_{t+k} = L_t + kT_t + S_{t-p+k}$ |
| **Level** | $L_t = \theta Y_t + (1-\theta) L_{t-1}$ |
| **Trend** | $T_t = \gamma(L_t - L_{t-1}) + (1-\gamma) T_{t-1}$ |
| **Season** | $S_t = \delta(Y_t - L_t) + (1-\delta) S_{t-p}$ |



# Stationarity

- Need consistency of mean and variance.
- What about changes in mean – trending, seasonality? **NOT** stationary.

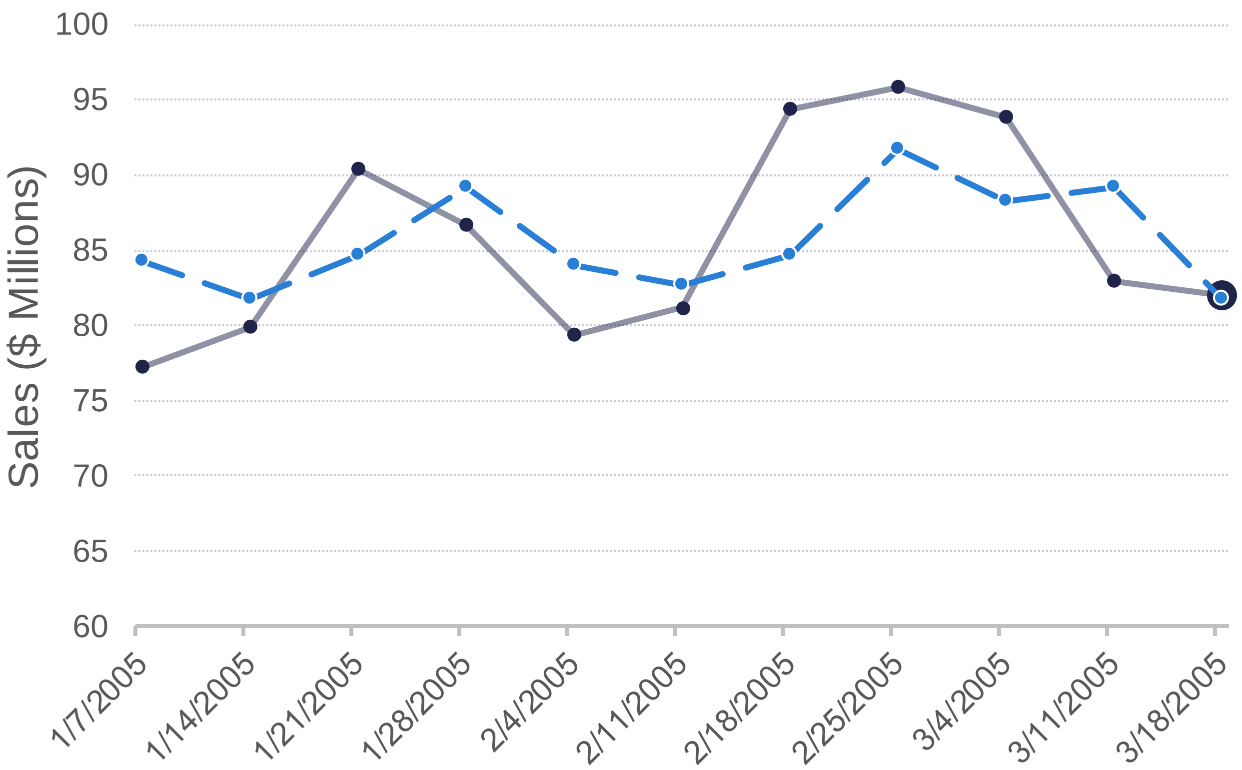


# ARIMA Models

- AR – forecast a series based solely on the past values in the series – called **lags.**

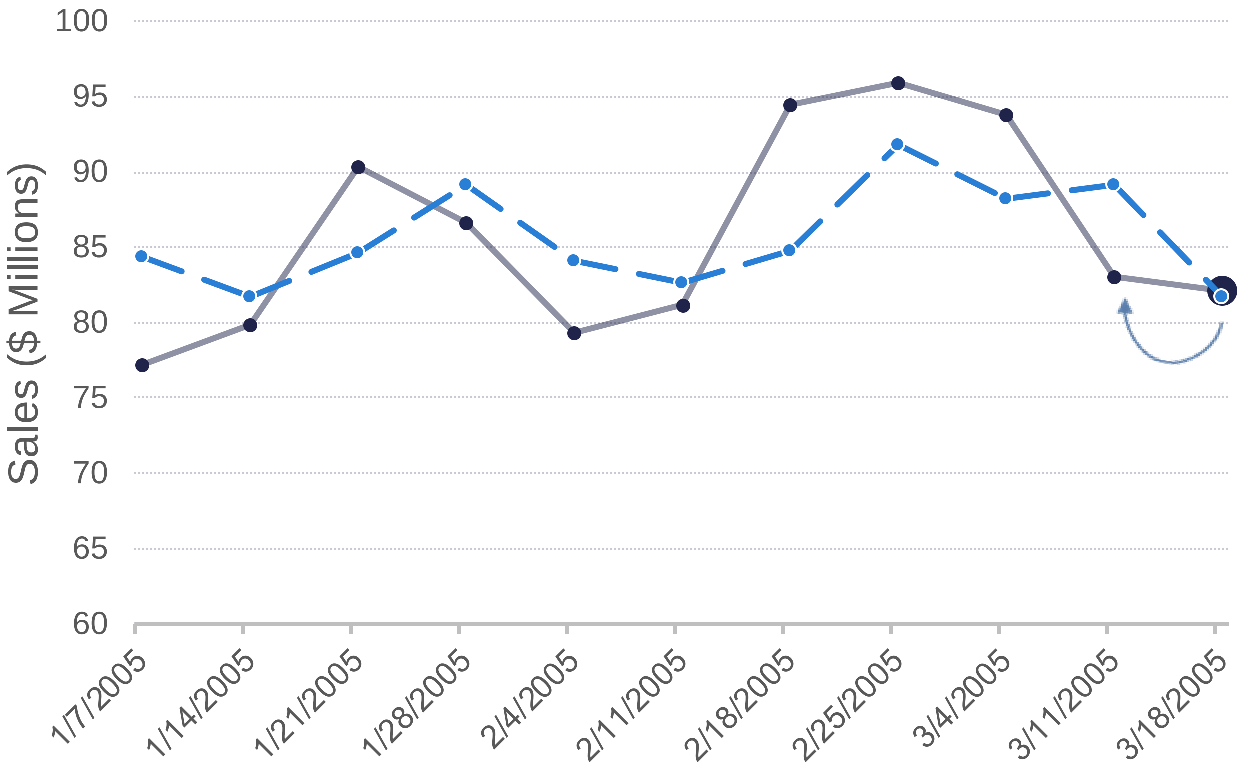
- MA – forecast a series based solely on the past errors in the series – called **error lags**.



# ARIMA Models

- AR – forecast a series based solely on the past values in the series – called **lags.**

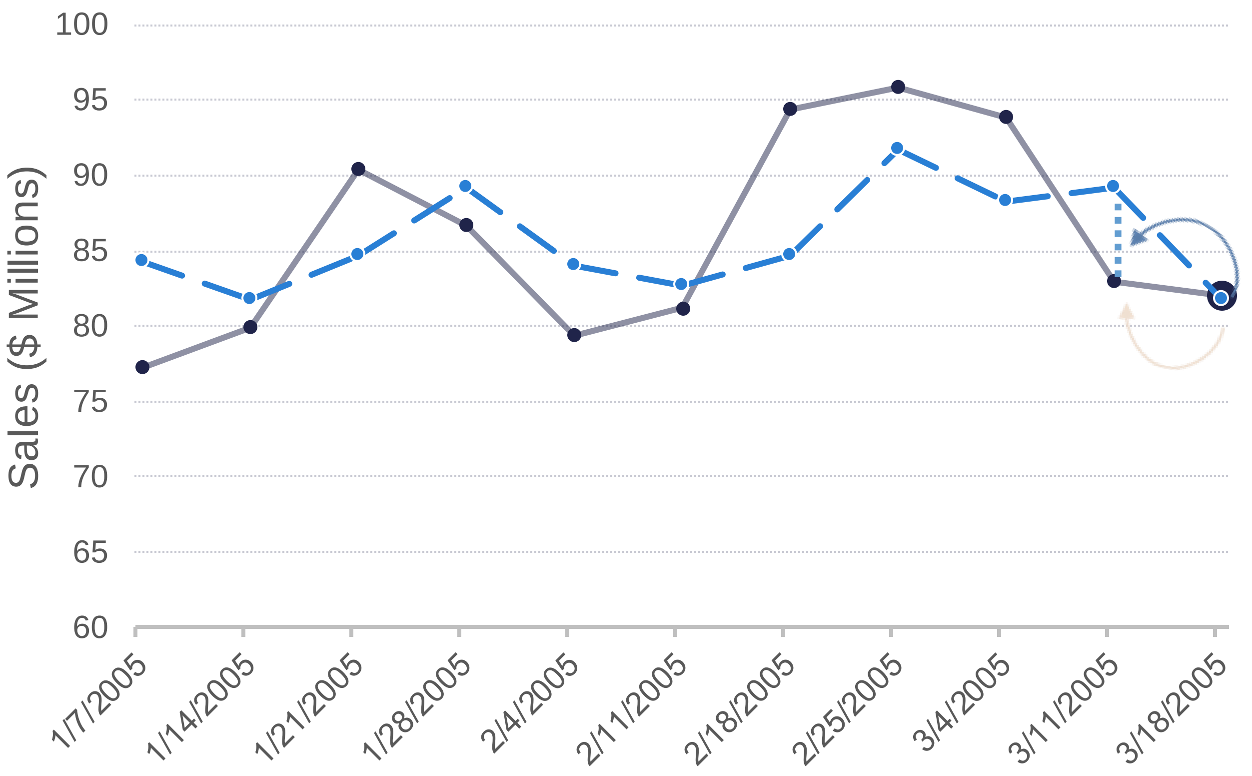
- MA – forecast a series based solely on the past errors in the series – called **error lags.**



# ARIMA Models

- AR – forecast a series based solely on the past values in the series – called **lags.**

- MA – forecast a series based solely on the past errors in the series – called **error lags**.



# ARIMA Models

- ARIMA Models are typically written as the following:

$$\text{ARIMA}(p, d, q)$$

# of AR Terms

# of First Differences

# of MA Terms

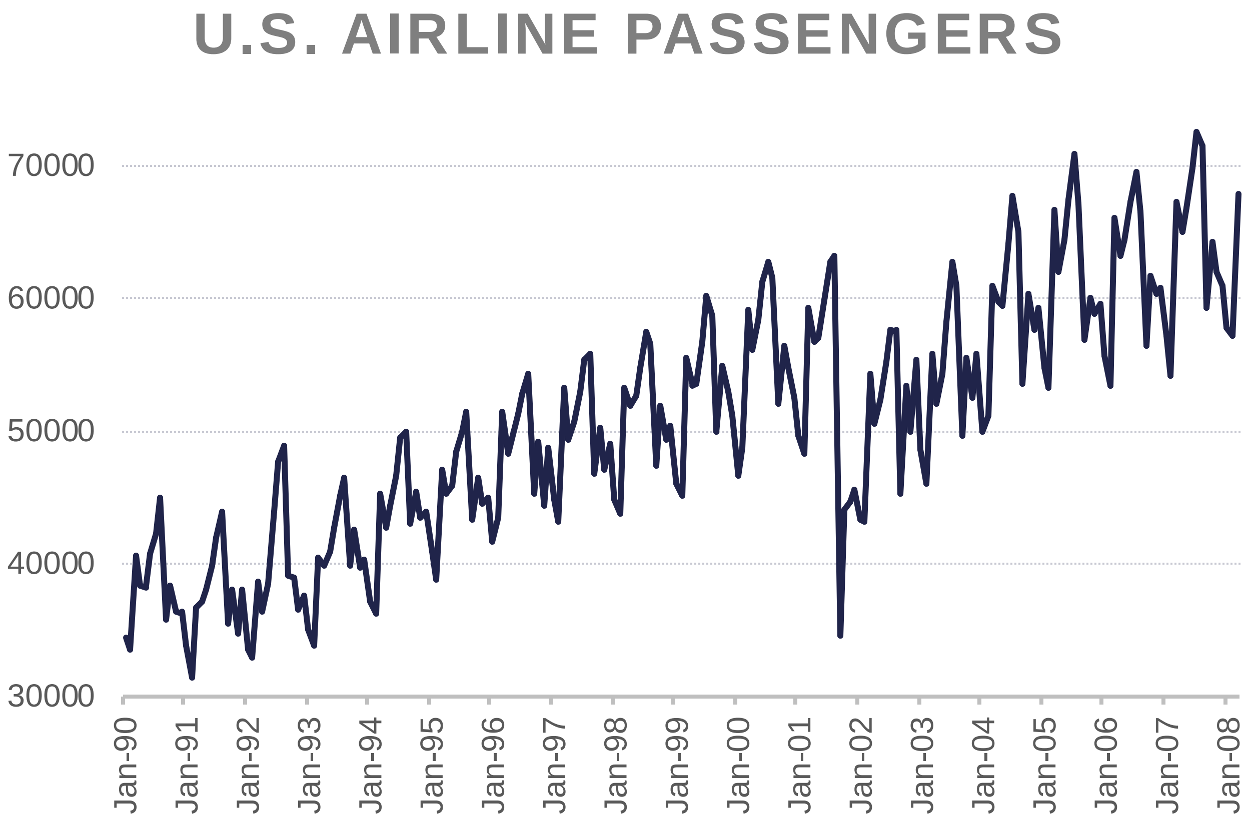
# U.S. Airlines Passengers 1990 – 2007

**Original Series**

**Trend / Cycle**

**Season**

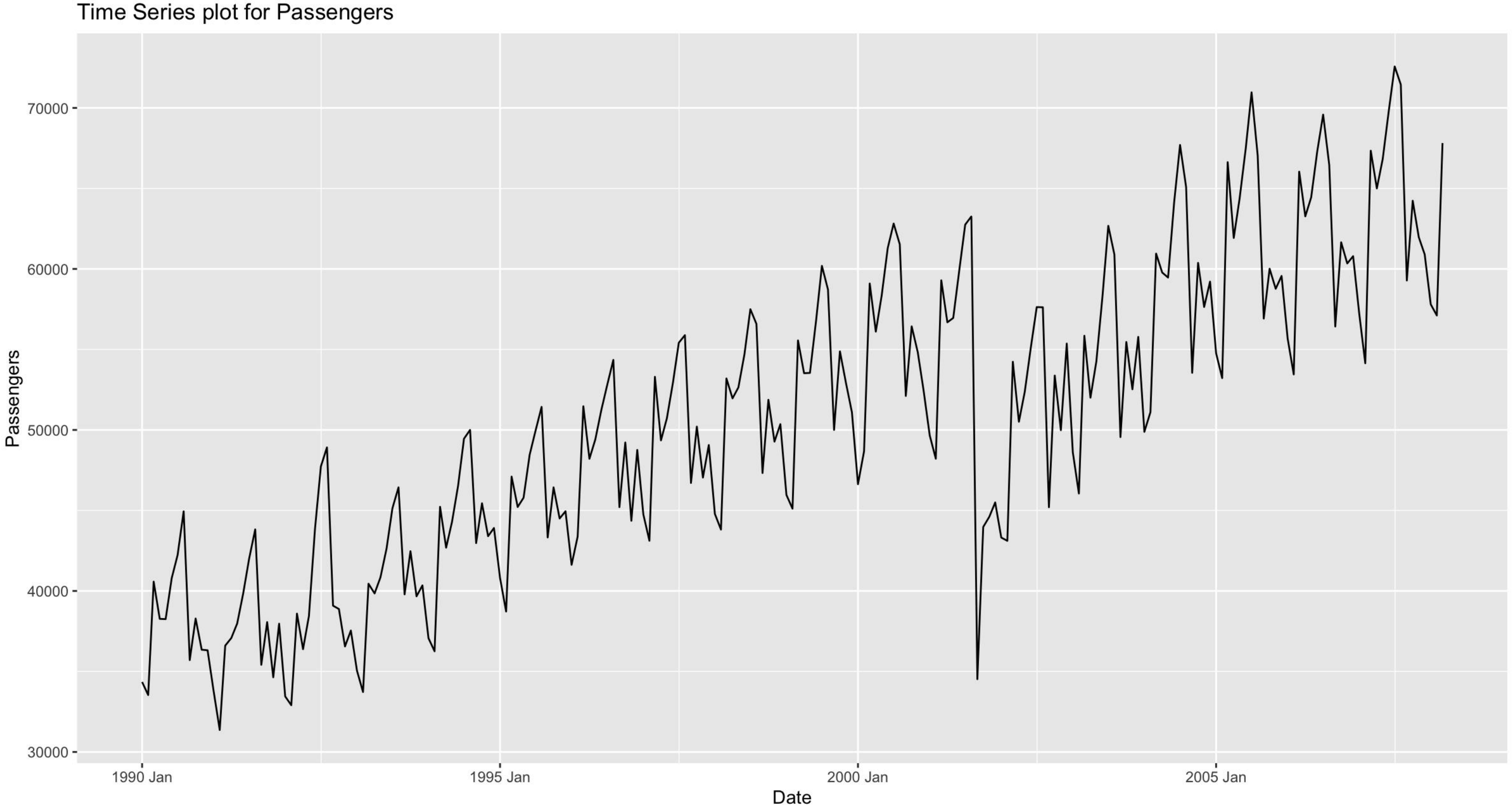
**Error**



# U.S. Airlines Passengers 1990 – 2007

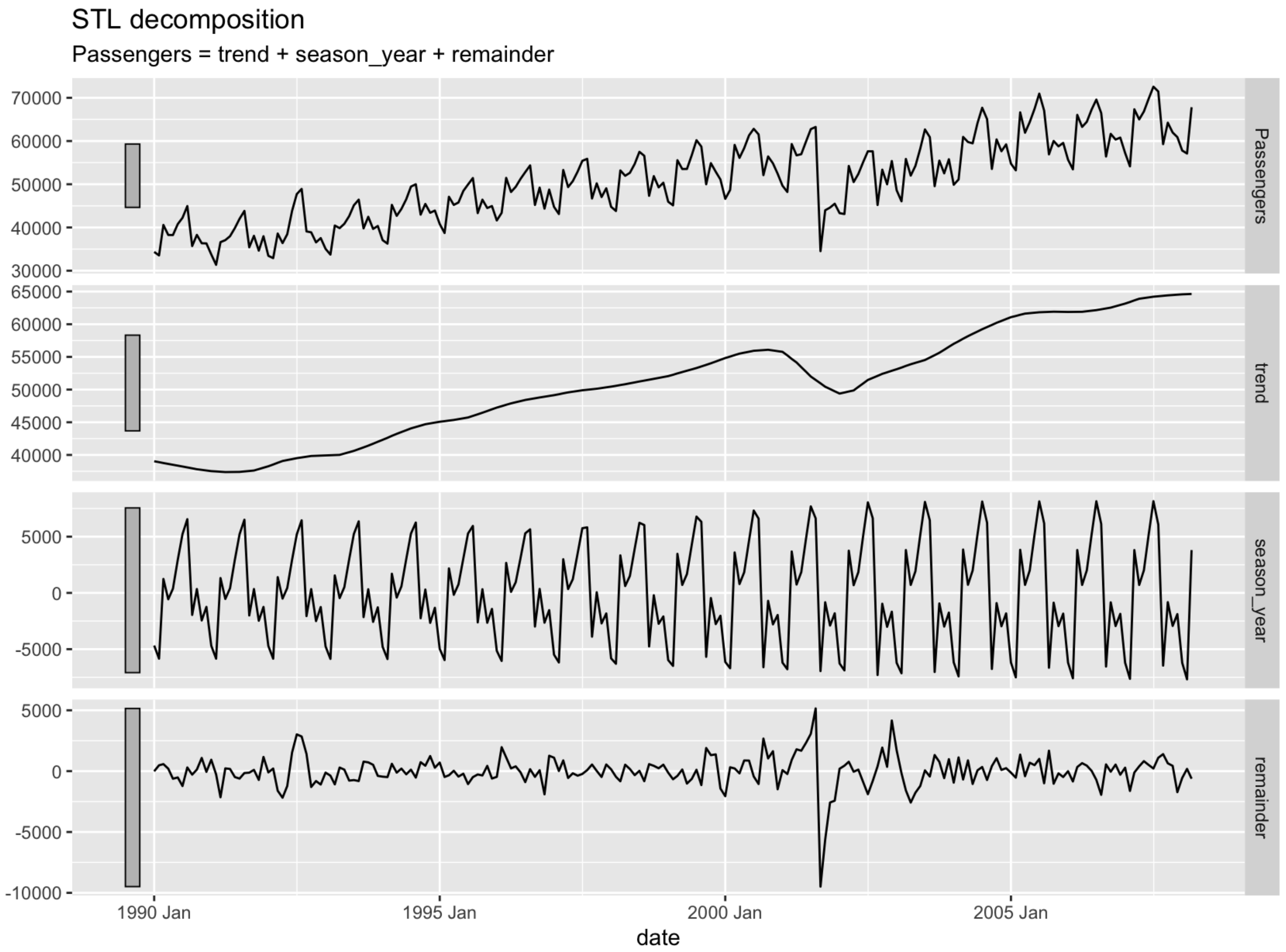
```
file.dir = "https://raw.githubusercontent.com/sjsimmo2/TimeSeries/master/"
input.file1 = "usairlines.csv"

USAirlines = read.csv(paste(file.dir, input.file1,sep = ""))

USAirlines <- USAirlines %>%
  mutate(date = yearmonth(lubridate::make_date(Year, Month)))

USAirlines_ts <- as_tsibble(USAirlines, index = date)

autoplot(USAirlines_ts, Passengers) +
  labs(title = "Time Series plot for Passengers", x = "Date", y = "Passengers")
```

# U.S. Airlines Passengers 1990 – 2007



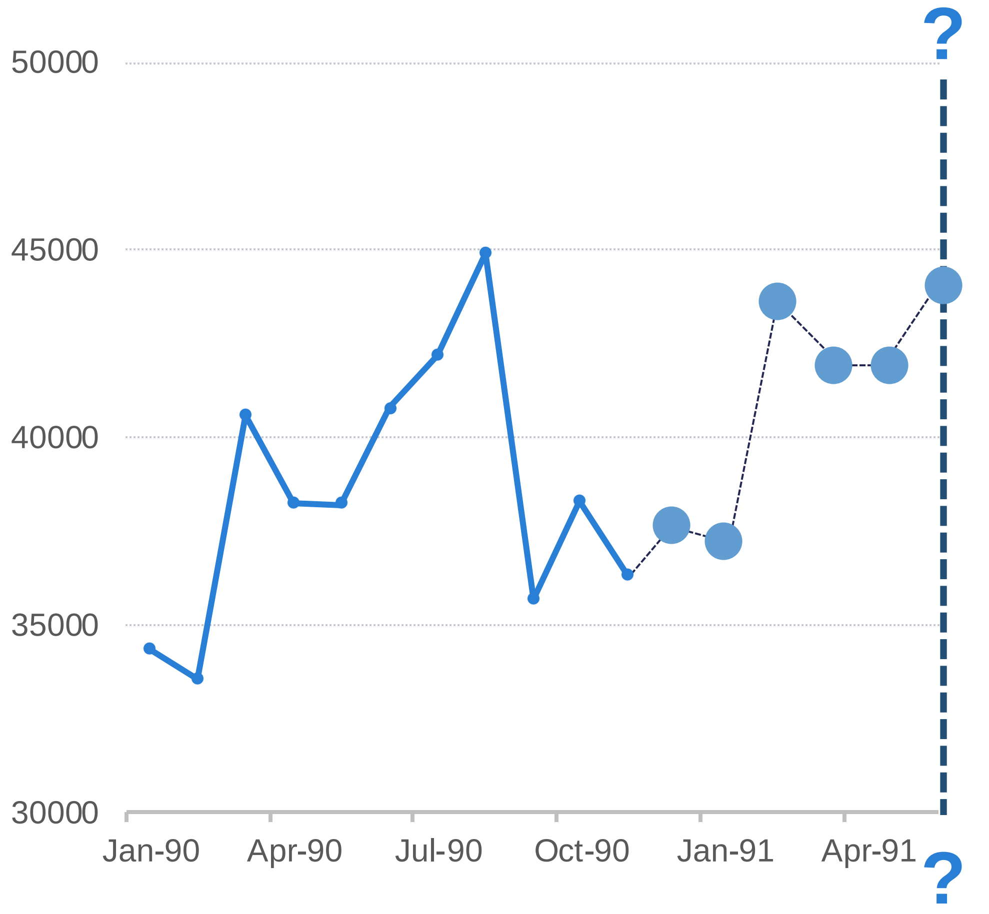
# Split into Training and Validation

```
train <- USAirlines_ts %>%
  select(Passengers, date, Month) %>%
  filter_index(~ "2007-03")

test <- USAirlines_ts %>%
  select(Passengers, date, Month) %>%
  filter_index("2007-04" ~ .)

dcmp <- USAirlines_ts %>%
  model(stl = STL(Passengers))

components(dcmp) %>% autoplot()
```



# Exponential Smoothing Models

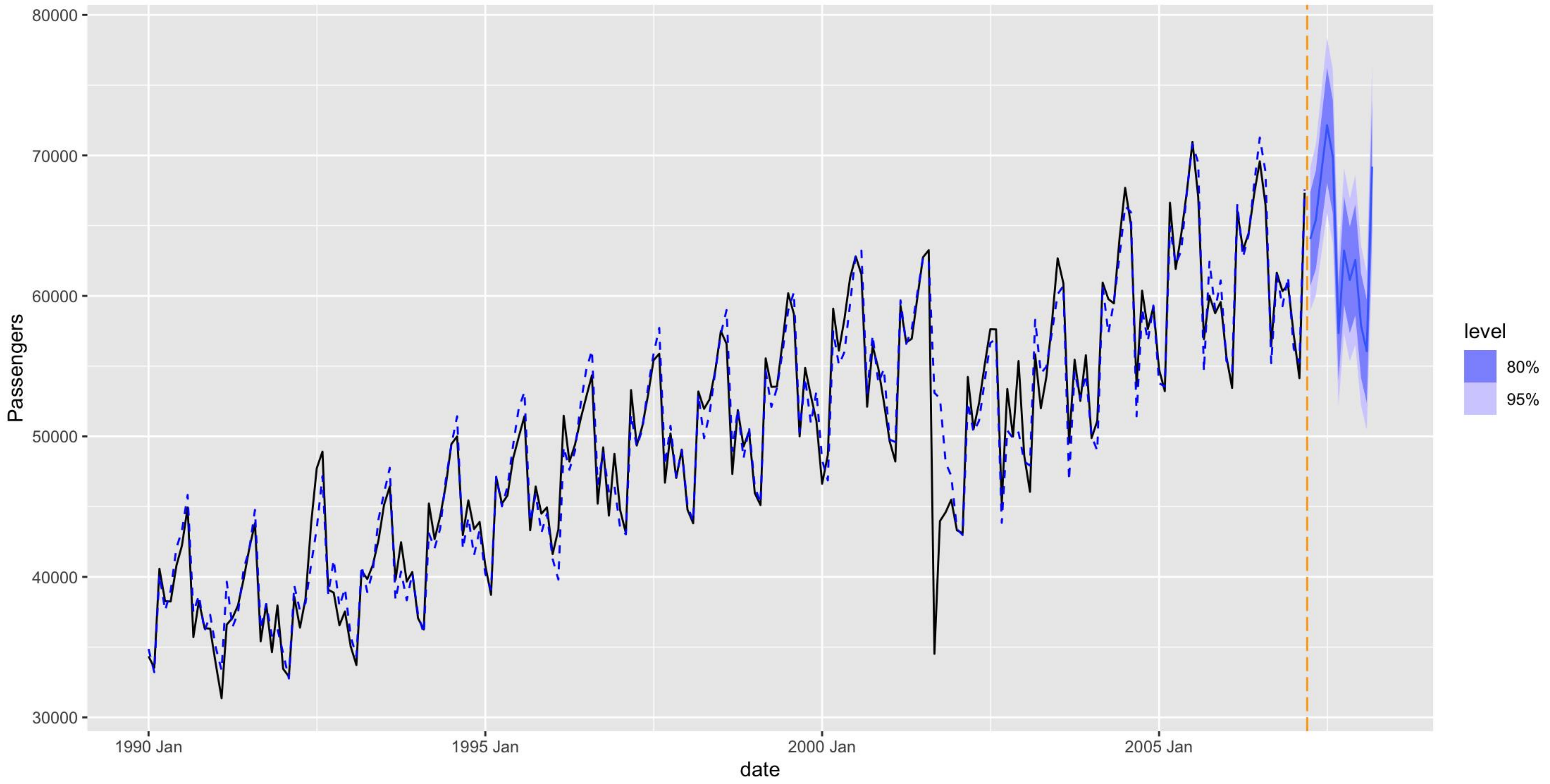
| | |
|---|---|
| **Original Series** | $\widehat{Y}_{t+k} = L_t + kT_t + S_{t-p+k}$ |
| = | = |
| **Level** | $L_t = \theta Y_t + (1-\theta)L_{t-1}$ |
| **Trend** | $T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1}$ |
| **Season** | $S_t = \delta(Y_t - L_t) + (1-\delta)S_{t-p}$ |



# Exponential Smoothing Models

```
model_HW <- train %>%
  model(
    ETS(Passengers ~ error("M") + trend("A") + season("M"))
  )

model_HW_for <- model_HW %>%
  fabletools::forecast(h = 12)

fabletools::accuracy(model_HW_for, test)
```

# Exponential Smoothing Models



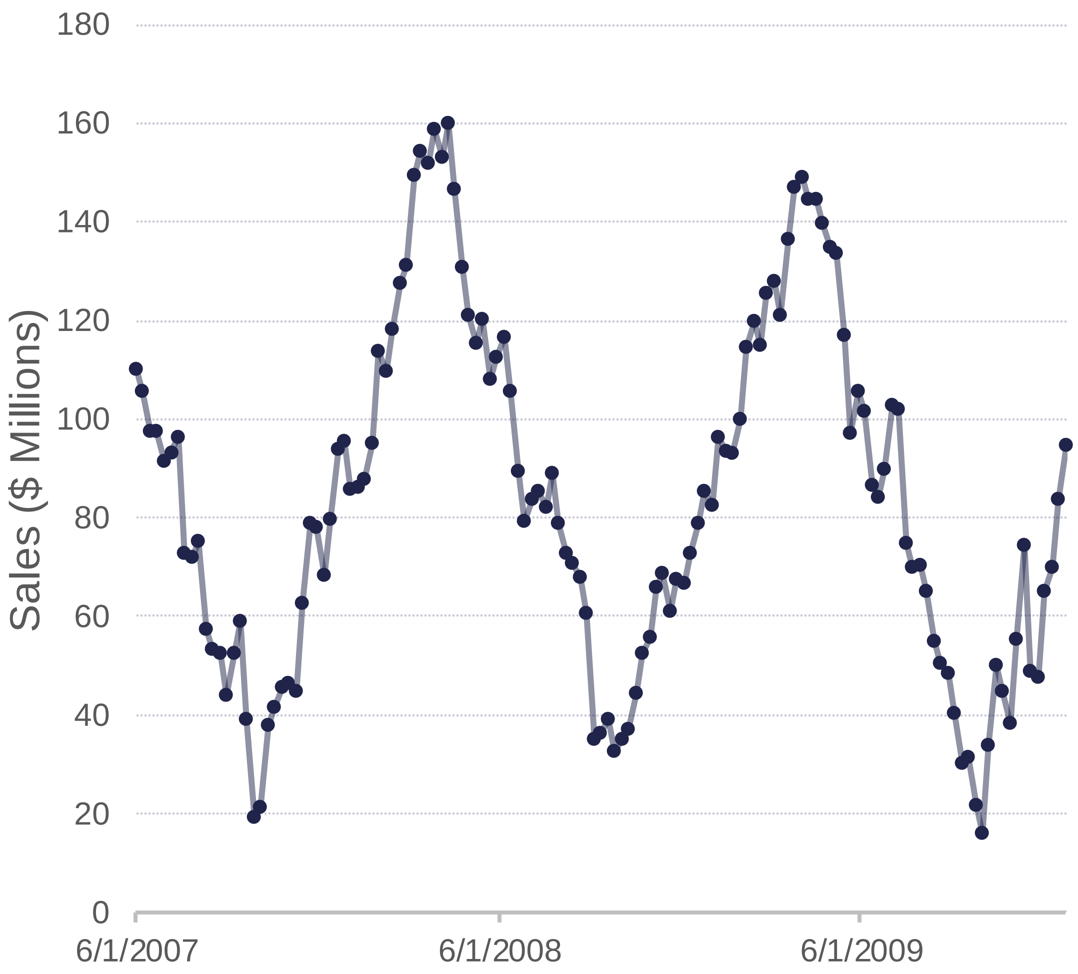
# Model Evaluation on Test Data

| Model | MAE | MAPE |
|---|---|---|
| HW Exponential Smoothing | 1100.02 | 1.71% |

# SEASONALITY

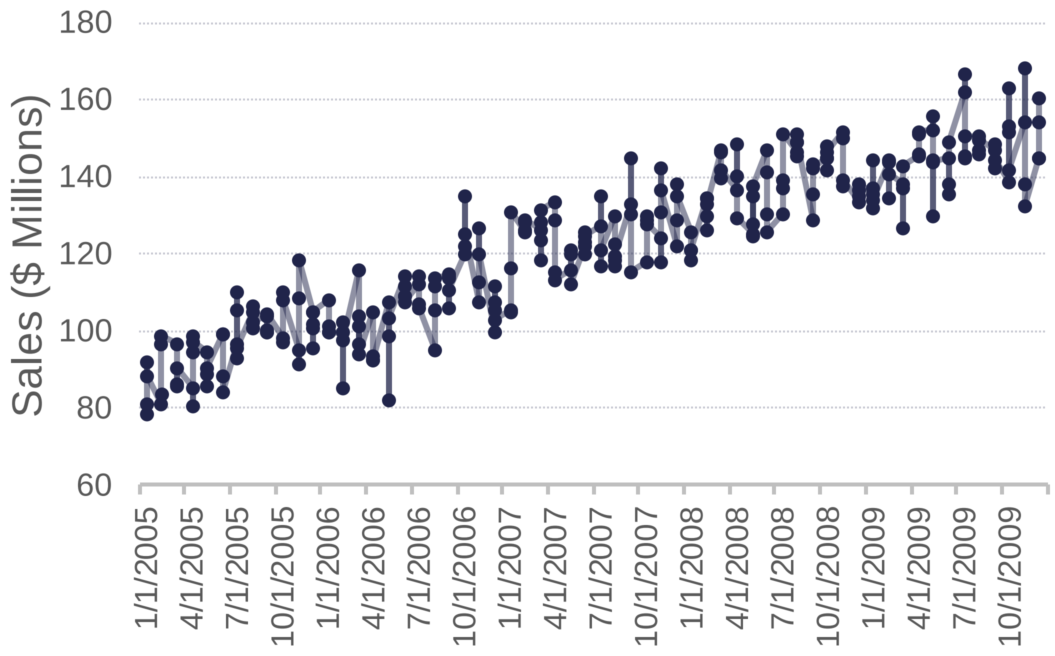
# Seasonality

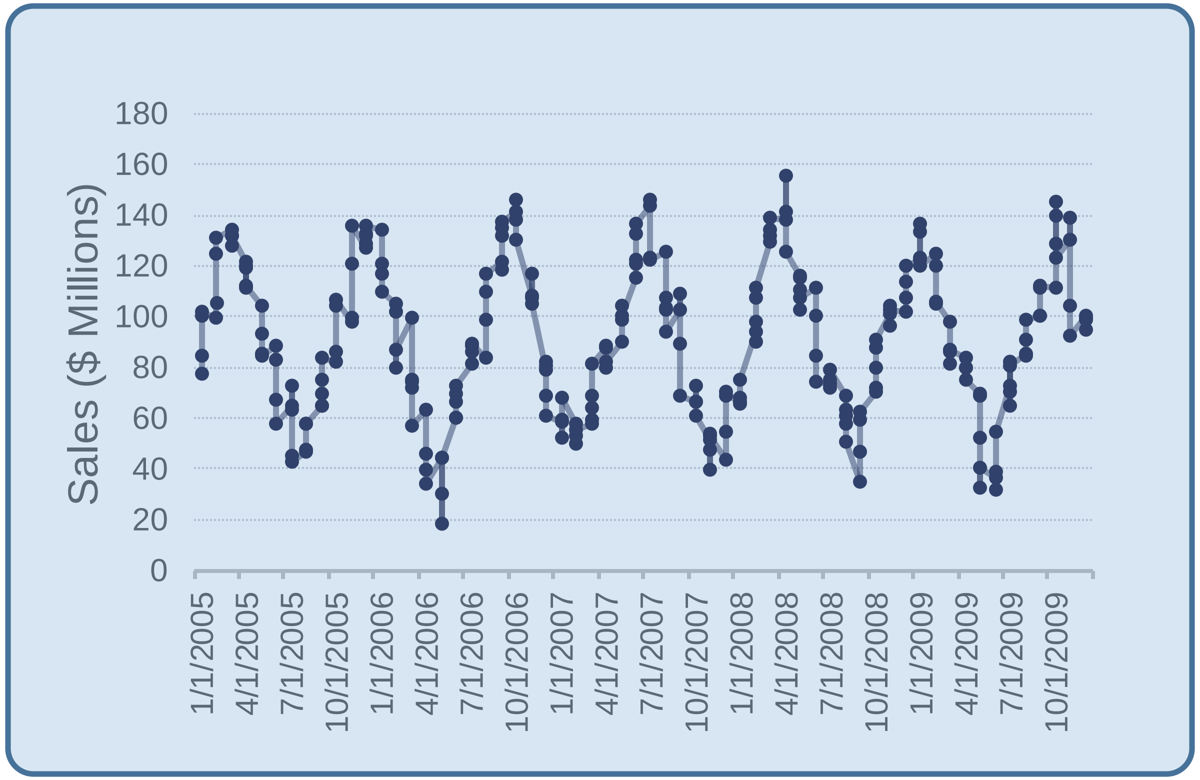
- Seasonality is the component of time series that represents the effects of seasonal variation.
- Component that describes repetitive behavior known as seasonal periods.
  - Seasonal period = S
  - Seasonal factors repeat every S units of time.



# Seasonality and Stationarity

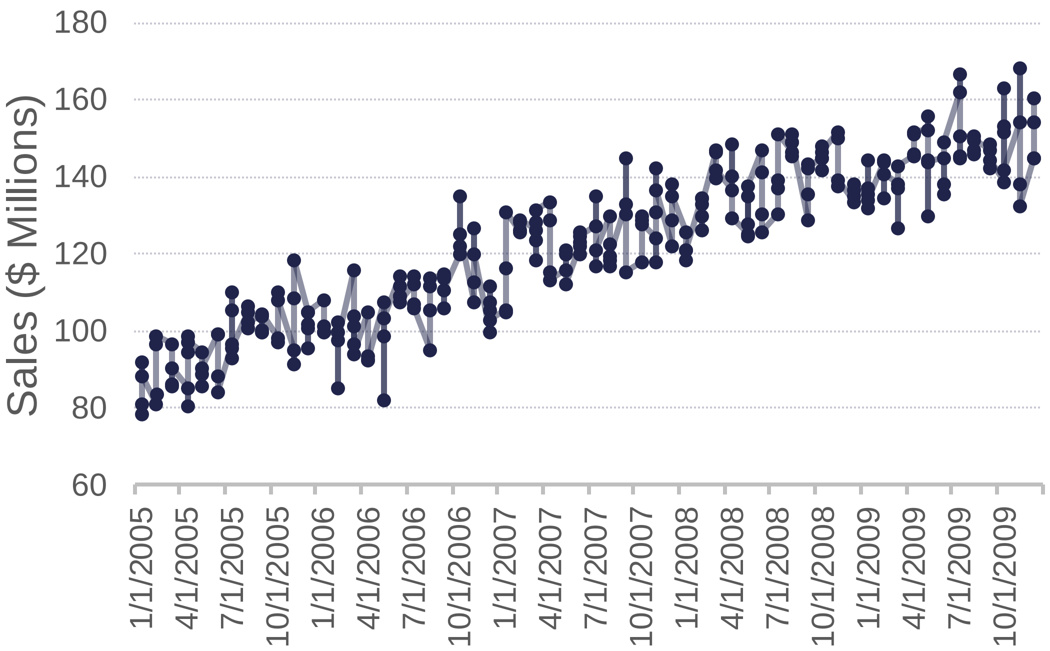
- Need consistency of mean and variance.
- What about changes in mean – trending, seasonality? **NOT** stationary.

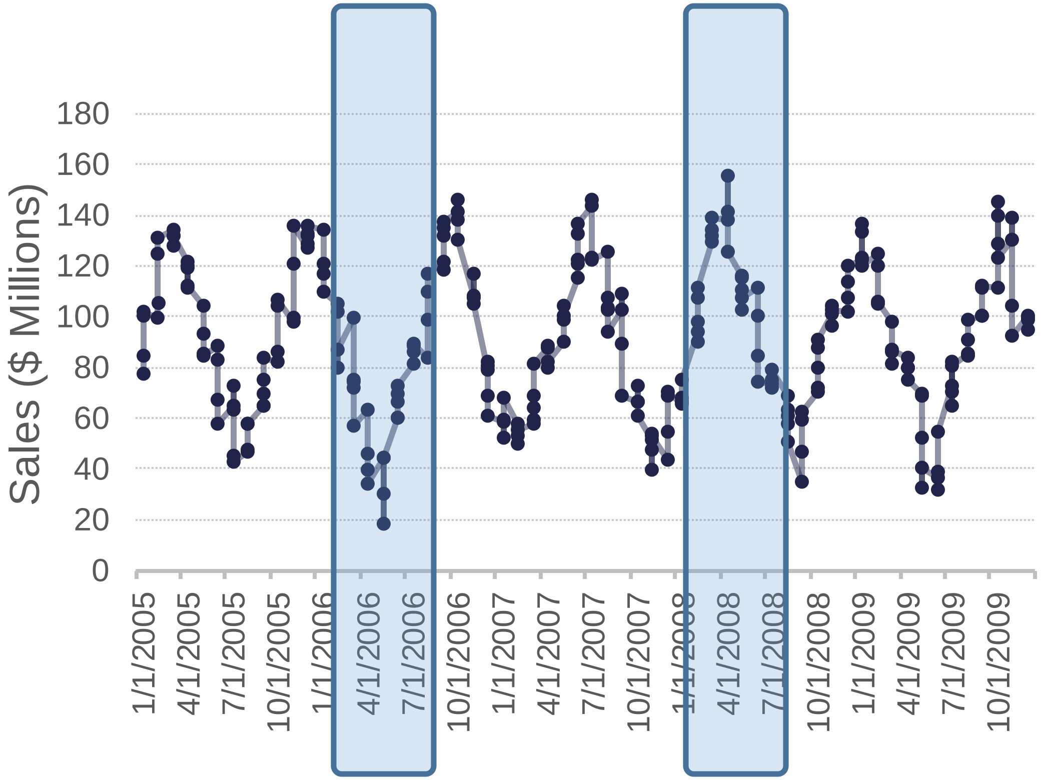


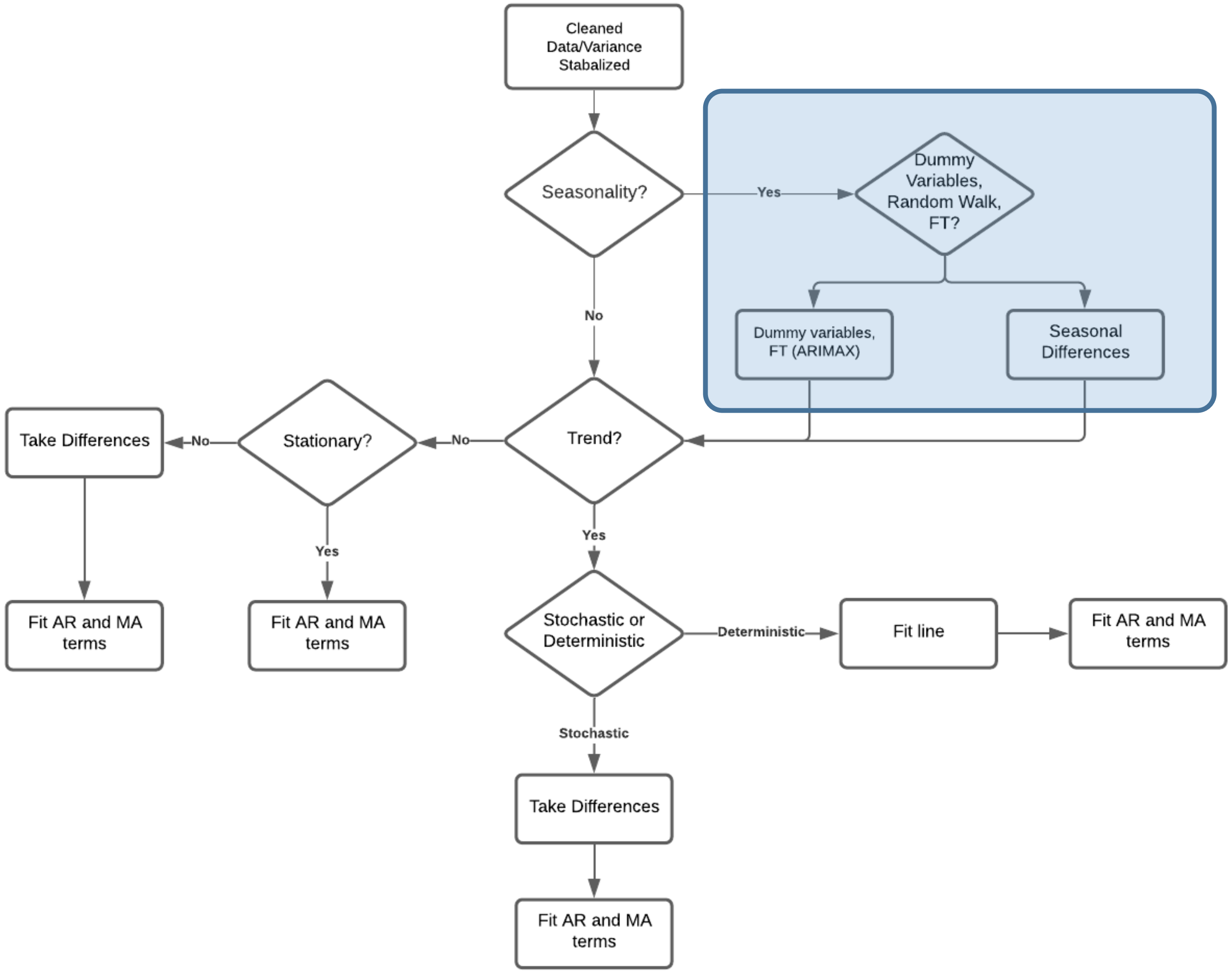
# Seasonality and Stationarity

- Need consistency of mean and variance.
- What about changes in mean – trending, seasonality? **NOT** stationary.



# ARIMA Framework



# Seasonal ARIMA Models

- Similar to trend, seasonality can be solved with a deterministic solution or a stochastic solution.
  - **Deterministic** – Seasonal dummy variables, Fourier transforms, predictor variables
  - **Stochastic** – Seasonal differences

- Once data is made stationary, we can model with traditional ARIMA approaches.
- Convert back to original for forecasting.

# Seasonal Unit-Root “Testing”

- Similar to trend, we can perform statistical tests for evaluating whether a unit root exists for seasonal data.
  - **Seasonal** unit root tests have problems with large seasonal frequencies – anything over 12 data points large for a season.

# Seasonal “Testing”

- Similar to trend, we can perform statistical tests for evaluating whether a unit root exists for seasonal data.
  - **Seasonal** unit root tests have problems with large seasonal frequencies – anything over 12 data points large for a season.

- To counter problems with seasonal unit root tests, can use measures of seasonal strength:

$$F_S = \max\left(0, 1 - \frac{Var(R_t)}{Var(S_t + R_t)}\right)$$

# Seasonal "Testing"

- Similar to trend, we can perform statistical tests for evaluating whether a unit root exists for seasonal data.
  - **Seasonal** unit root tests have problems with large seasonal frequencies – anything over 12 data points large for a season.

- To counter problems with seasonal unit root tests, can use measures of seasonal strength:

$$F_S = \max\left(0, 1 - \frac{Var(R_t)}{Var(S_t + R_t)}\right)$$

Residual (Error) component

Seasonal component

# Seasonal "Testing"

- Similar to trend, we can perform statistical tests for evaluating whether a unit root exists for seasonal data.
  - **Seasonal** unit root tests have problems with large seasonal frequencies – anything over 12 data points large for a season.

- To counter problems with seasonal unit root tests, can use measures of seasonal strength:

$$F_S = \max\left(0, 1 - \frac{Var(R_t)}{Var(S_t + R_t)}\right)$$

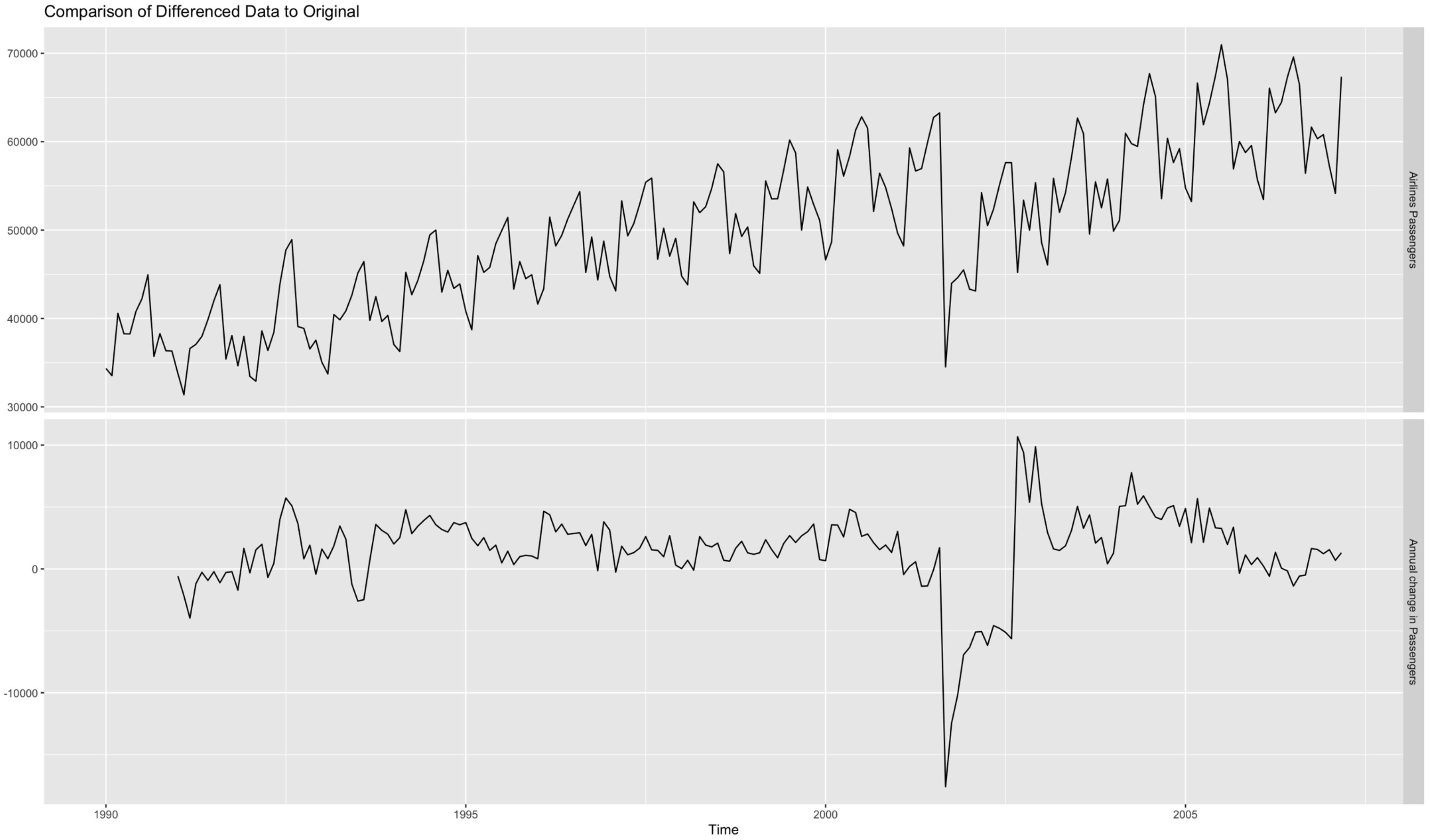
If $F_S < 0.64$ then no seasonal differencing, otherwise 1 seasonal difference

# Seasonal “Testing”

```
train %>%
   features(Passengers, unitroot_nsdiffs)

# A tibble: 1 × 1
  nsdiffs
   <int>
1       1
```

Should take one **seasonal** difference

# Differenced Data



# Unit-Root Testing

```
train %>%
  mutate(Pass_diff = difference(Passengers, lag = 12)) %>%
  features(Pass_diff, unitroot_ndiffs)

# A tibble: 1 × 1
  ndiffs
   <int>
1      0
```

Should take 0 **regular** differences AFTER taking the seasonal difference

# DETERMINISTIC SOLUTIONS

# Which Deterministic Solution?

- Similar to trend, seasonality can be solved with a deterministic solution or a stochastic solution.
  - **Deterministic** – Seasonal dummy variables, Fourier transforms, predictor variables
  - **Stochastic** – Seasonal differences

- Once data is made stationary (model away the seasonality), we can model with traditional ARIMA approaches.

# Seasonal Dummy Variables

- For a time series with *S* periods within a season, there will be *S-1* dummy variables, one for each period (and one accounted for with the intercept).
- Monthly Data:
  - One dummy variable for each month (S = 12)
- Weekly Data:
  - One dummy variable for each day of week (S = 7)
- Hourly Data:
  - One dummy variable for each hour (S = 24)

# Seasonal Dummy Variables

- Example model with intercept:

$$Y_t = \beta_0 + \beta_1 JAN + \beta_2 FEB + \cdots + \beta_{11} NOV + e_t$$

$$\beta_0 + \beta_M = \text{effect of M}^{\text{th}} \text{ month}$$

$$\beta_0 = \text{effect of December}$$

# Seasonal Dummy Variables

```
season_lin <- lm(Passengers ~ factor(Month), data = train)

summary(season_lin)


Coefficients:

                Estimate Std. Error t value Pr(>|t|)

(Intercept)      44299.6     1945.9  22.766  < 2e-16 ***
factor(Month)2    -971.9     2751.9  -0.353  0.72433
factor(Month)3    8561.7     2751.9   3.111  0.00214 **
factor(Month)4    5276.9     2792.1   1.890  0.06025 .
factor(Month)5    6413.7     2792.1   2.297  0.02267 *
factor(Month)6    9306.5     2792.1   3.333  0.00103 **
factor(Month)7   11966.8     2792.1   4.286 2.86e-05 ***
factor(Month)8   11759.3     2792.1   4.212 3.87e-05 ***
factor(Month)9    1219.4     2792.1   0.437  0.66279
factor(Month)10   5526.0     2792.1   1.979  0.04920 *
factor(Month)11   3193.7     2792.1   1.144  0.25408
factor(Month)12   4461.9     2792.1   1.598  0.11165

---
```

# Seasonal Dummy Variables

```
model_SD_ARIMA <- train %>%
  model(ARIMA(Passengers ~ factor(Month) + PDQ(0,0,0)))
report(model_SD_ARIMA)
```

# Seasonal Dummy Variables

```
Series: Passengers
Model: LM w/ ARIMA(1,1,1) errors


Coefficients:
         ar1      ma1  factor(Month)2  factor(Month)3  factor(Month)4
      0.4290  -0.7970      -1092.9154       8320.1242       5723.4668
s.e.  0.1142   0.0773        473.9114        574.5485        625.8415
      factor(Month)5  factor(Month)6  factor(Month)7  factor(Month)8
           6721.1774       9485.2351      12021.9750      11693.0530
s.e.        648.7665        659.2791        662.5646        659.9506
      factor(Month)9  factor(Month)10  factor(Month)11  factor(Month)12  intercept
           1033.6595        5223.6760        2779.6623        3948.1224   120.7069
s.e.        650.4285         629.7997         586.4674         485.1739    47.0971


sigma^2 estimated as 3751115:  log likelihood=-1844.41
AIC=3718.82   AICc=3721.34   BIC=3768.74
```

# Advantages and Disadvantages

## Advantages

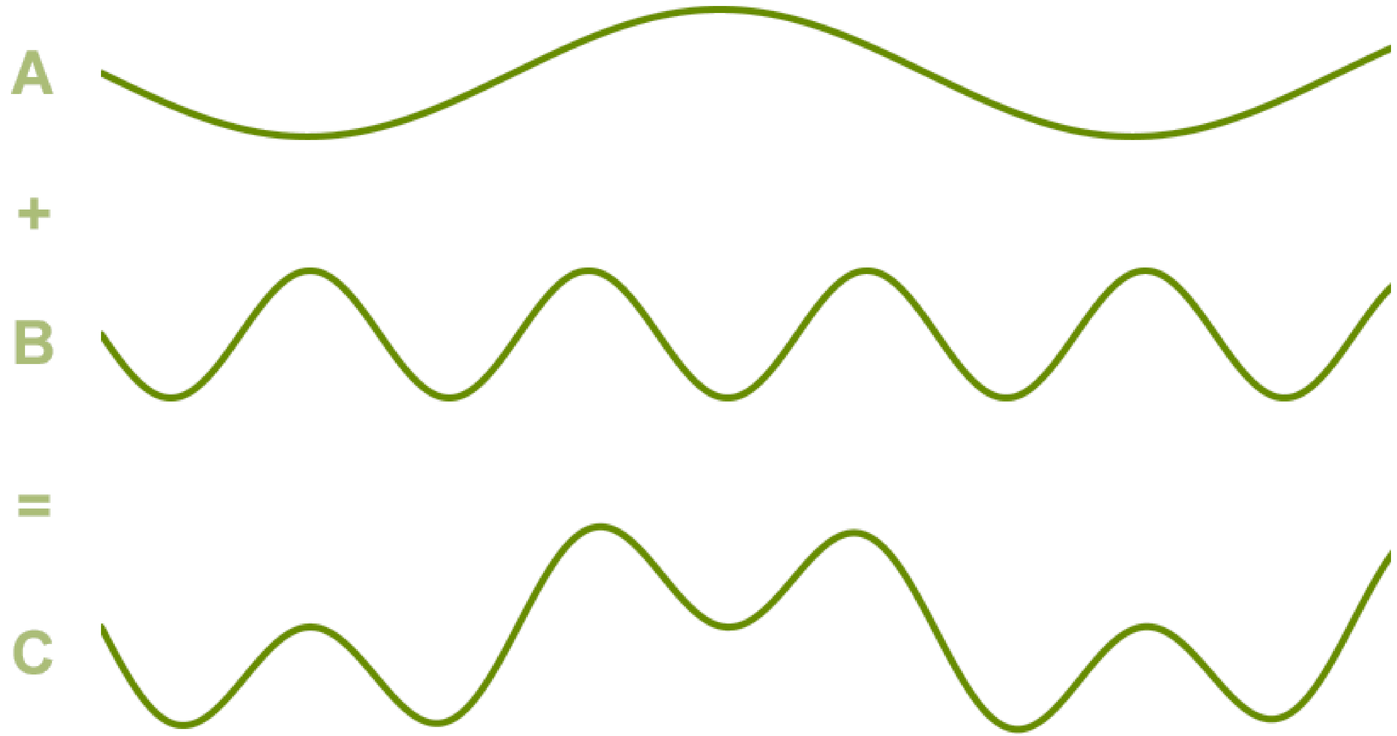
- Interpretation still holds.
  - Can easily measure and interpret effects from different parts of the season.
- Straight forward to implement.

## Disadvantages

- Especially long or complex seasons are hard to deal with.
  - More than 24 periods in a season (365 days in year for example) is burdensome.
  - Some seasons are complex (365.25 days in a year, 52.17 weeks in a year, etc.).
- Seasonal effects remain constant.

# Fourier Transforms (Harmonic Regression)

- Fourier showed that series of sine and cosine terms of the right frequencies approximate periodic series.



# Fourier Transforms (Harmonic Regression)

- Add Fourier variables to a regression model predicting the target to remove the seasonal pattern.

$$X_{1,t} = \sin\left(\frac{2\pi t}{S}\right) \qquad X_{3,t} = \sin\left(2 \times \frac{2\pi t}{S}\right) \qquad X_{5,t} = \sin\left(3 \times \frac{2\pi t}{S}\right) \qquad \cdots$$

$$X_{2,t} = \cos\left(\frac{2\pi t}{S}\right) \qquad X_{4,t} = \cos\left(2 \times \frac{2\pi t}{S}\right) \qquad X_{6,t} = \cos\left(3 \times \frac{2\pi t}{S}\right) \qquad \cdots$$

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \beta_3 X_{3,t} + \beta_4 X_{4,t} + \cdots + e_t$$

# Fourier Transforms (Harmonic Regression)

- Add Fourier variables to a regression model predicting the target to remove the seasonal pattern.
- If you add the same number of Fourier variables as you have seasonal dummy variables, you will get the same predictions.
- However, typically do not need all the Fourier variables → especially with large values of *S*.

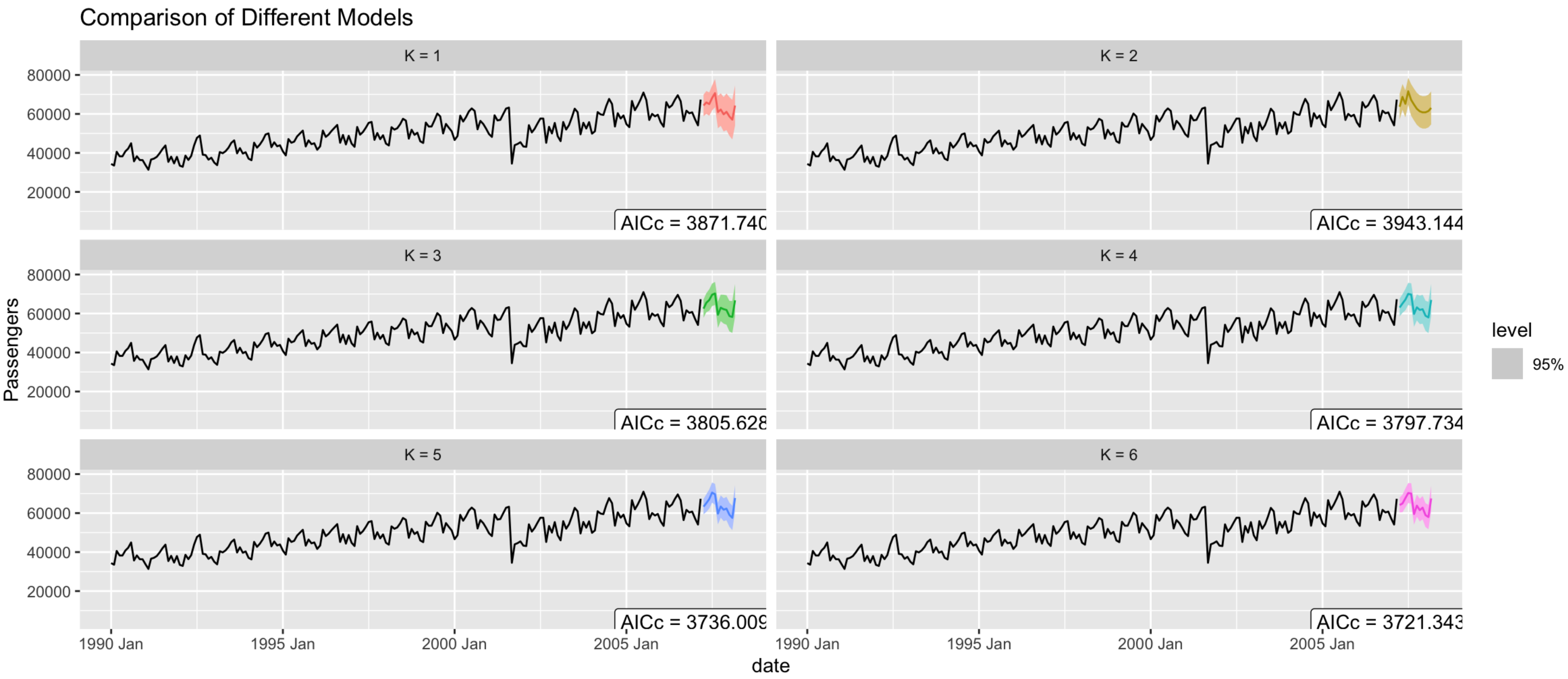
# Fourier Transforms (Harmonic Regression)

```
model_F_ARIMA <- train %>%
  model(
  `K = 1` = ARIMA(Passengers ~ fourier(K = 1) + PDQ(0,0,0)),
  `K = 2` = ARIMA(Passengers ~ fourier(K = 2) + PDQ(0,0,0)),
  `K = 3` = ARIMA(Passengers ~ fourier(K = 3) + PDQ(0,0,0)),
  `K = 4` = ARIMA(Passengers ~ fourier(K = 4) + PDQ(0,0,0)),
  `K = 5` = ARIMA(Passengers ~ fourier(K = 5) + PDQ(0,0,0)),
  `K = 6` = ARIMA(Passengers ~ fourier(K = 6) + PDQ(0,0,0))
)

glance(model_F_ARIMA)
```

# Fourier Transforms (Harmonic Regression)

```
# A tibble: 6 × 8
  .model     sigma2 log_lik   AIC  AICc   BIC ar_roots   ma_roots
  <chr>       <dbl>   <dbl> <dbl> <dbl> <dbl> <list>     <list>

1 K = 1   7934443.  -1926. 3871. 3872. 3901. <cpl [5]> <cpl [1]>
2 K = 2  11151798.  -1960. 3942. 3943. 3978. <cpl [0]> <cpl [5]>
3 K = 3   5670629.  -1889. 3804. 3806. 3847. <cpl [4]> <cpl [1]>
4 K = 4   5388483.  -1883. 3795. 3798. 3845. <cpl [4]> <cpl [1]>
5 K = 5   4027199.  -1852. 3733. 3736. 3783. <cpl [2]> <cpl [1]>
6 K = 6   3751119.  -1844. 3719. 3721. 3769. <cpl [1]> <cpl [1]>
```

# Fourier Transforms (Harmonic Regression)



# Fourier Transforms (Harmonic Regression)

```
model_F_ARIMA <- train %>%
  model(ARIMA(Passengers ~ fourier(K = 6) + PDQ(0,0,0))
)

report(model_F_ARIMA)
```

# Fourier Transforms (Harmonic Regression)

```
Model: LM w/ ARIMA(1,1,1) errors

Coefficients:
         ar1      ma1  fourier(K = 6)C1_12  fourier(K = 6)S1_12
      0.4290  -0.7970           -4370.2035            1100.2519
s.e.  0.1142   0.0773             270.2329             270.4442
      fourier(K = 6)C2_12  fourier(K = 6)S2_12  fourier(K = 6)C3_12
                 610.7831            -430.1257           -2561.0814
s.e.             193.6263             194.1363             152.7946
      fourier(K = 6)S3_12  fourier(K = 6)C4_12  fourier(K = 6)S4_12
               -1291.4692             254.1196            -387.6305
s.e.             153.1538             130.6594             130.4205
      fourier(K = 6)C5_12  fourier(K = 6)S5_12  fourier(K = 6)C6_12
                 919.7895           -2141.1865            -341.9420
s.e.             118.9847             119.5054              81.8771
      intercept
       120.7034
s.e.    47.1035

sigma^2 estimated as 3751119:  log likelihood=-1844.41
AIC=3718.82   AICc=3721.34   BIC=3768.74
```

# Advantages and Disadvantages

### Advantages

- Can handle long and complex seasonality.
  - If multiple seasons, just add more Fourier variables to account for them.

### Disadvantages

- Trial and error for “right” amount of Fourier variables to use.
- No interpretable value.
- Effect of season remains constant.

# Predictor Variables for Seasonality

- Last common approach to accounting for seasonality in data is to use other predictor variables that have matching season.
- Modeling these variables against the target might remove the seasonality.
- Example: Weather data and energy data
  - Hourly temperature correlates with hourly energy usage in the summer months (high heat → high energy usage)
  - Have same 24-hour cycle

# Advantages and Disadvantages

## Advantages

- Can handle long and complex seasonality.
  - If multiple seasons, just add more variables to account for them.
- Interpretation still holds.
  - Can easily measure and interpret effects from these variables.

## Disadvantages

- Trial and error for “right” variables to use.
- Might not have predictor variables to use in this context.

# What Next?

- After removing the seasonality through deterministic approaches, the remaining error term (residuals) are modeled with Seasonal ARIMA models.

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \beta_3 X_{3,t} + \beta_4 X_{4,t} + \cdots + e_t$$

Seasonal ARIMA here!

- Still might need seasonal effects even though season is removed.

# STOCHASTIC SOLUTION (DIFFERENCING)

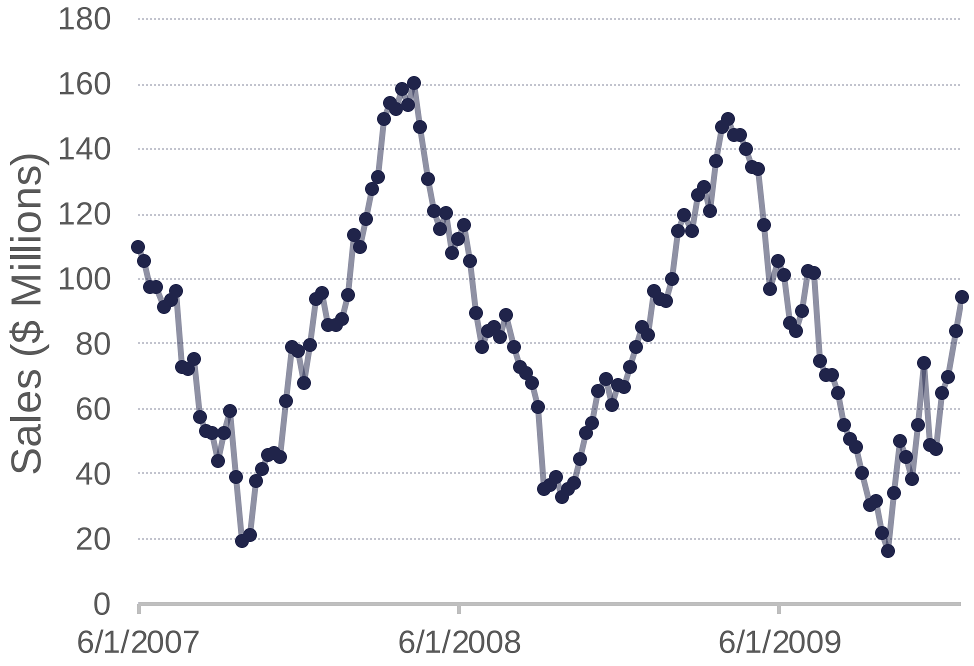
# Stochastic Solution

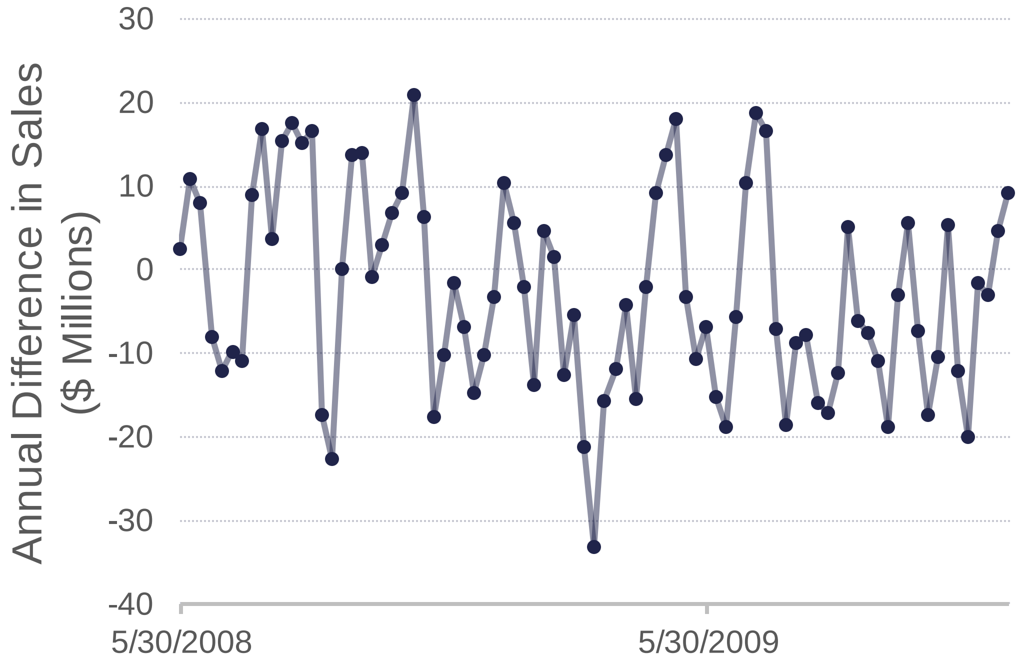
- Similar to trend, seasonality can be solved with a deterministic solution or a stochastic solution.
  - **Deterministic** – Seasonal dummy variables, Fourier transforms, predictor variables
  - **Stochastic** – Seasonal differences

- Once data is made stationary (model away the seasonality), we can model with traditional ARIMA approaches.

# Seasonal Differencing

- Differencing on season → look at difference between current point and the same point in the previous season: $Y_t - Y_{t-S}$



# What Next?

- After removing the seasonality through stochastic approaches, the remaining differences are modeled with Seasonal ARIMA models.

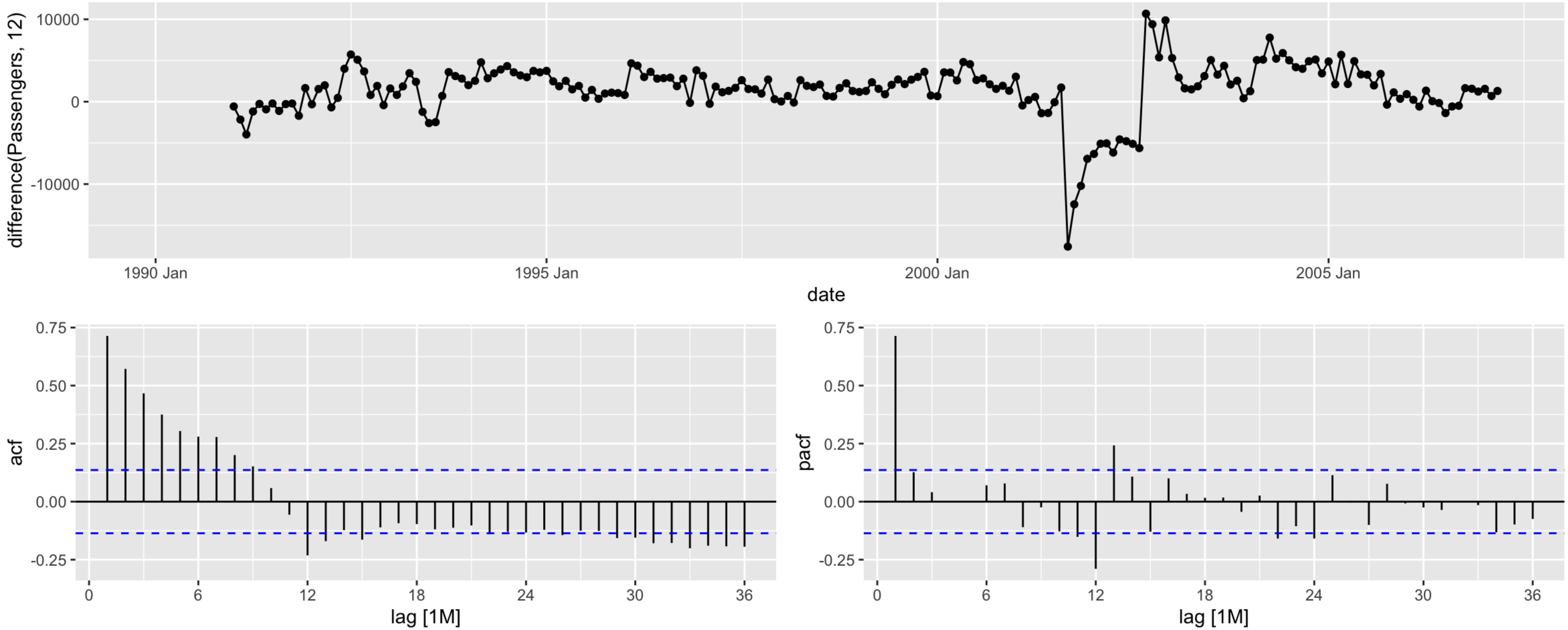
$$Y_t - Y_{t-s} = W_t$$

Seasonal ARIMA here!

- Still might need seasonal effects even though season is removed.

# Seasonal Differencing

```
train %>%
  gg_tsdisplay(difference(Passengers, 12), plot_type = 'partial', lag = 36)
```



# Limitations of Differencing

- Hard to evaluate stochastic effects for long and complex seasons.
- Most statistical tests for stochastic vs. deterministic can not handle past 12 or 24 periods in a season → need to use seasonal strength tests.
- Long/complex seasons → Best to just approach with deterministic solutions.

# SEASONAL ARIMA

# More Complex ARIMA

- When extending to the Seasonal ARIMA framework, we add another set of terms – *P*, *D*, *Q*, and *S*.

$$\text{ARIMA}(p, d, q)(P, D, Q)_S$$

**# of AR Terms**

**# of MA Terms**

**# of First Differences**

# More Complex ARIMA

- When extending to the Seasonal ARIMA framework, we add another set of terms – *P*, *D*, *Q*, and *S*.

$$\mathrm{ARIMA}(p,d,q)(P,D,Q)_S$$

**Length of Season**

**# of Seasonal AR Terms**

**# of Seasonal Differences**

**# of Seasonal MA Terms**

# Seasonal ARIMA

- **Seasonal** ARIMA models are typically written as the following:

$$\mathrm{ARIMA}(1,0,1)(2,1,0)_{12}$$

# Seasonal ARIMA

- **Seasonal** ARIMA models are typically written as the following:

$$\text{ARIMA}(1,0,1)(2,1,0)_{12}$$

$$Y_t - Y_{t-12} = W_t$$

# Seasonal ARIMA

- **Seasonal** ARIMA models are typically written as the following:

$$\text{ARIMA}(1,0,1)(2,1,0)_{12}$$

$$Y_t - Y_{t-12} = W_t$$

# Seasonal ARIMA

- **Seasonal** ARIMA models are typically written as the following:

$$\text{ARIMA}(1,0,1)(2,1,0)_{12}$$

$$Y_t - Y_{t-12} = W_t$$

$$W_t = \omega + \phi_1 W_{t-1} + \phi_2 W_{t-12} + \phi_3 W_{t-24} + \theta_1 e_{t-1} + e_t$$

# Seasonal ARIMA

- **Seasonal** ARIMA models are typically written as the following:

$$\text{ARIMA}(1,0,1)(2,1,0)_{12}$$

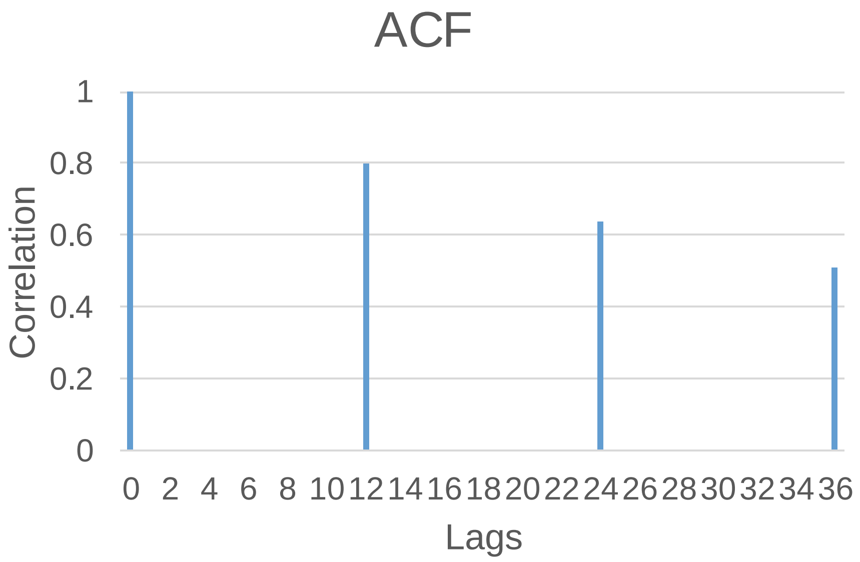
$$Y_t - Y_{t-12} = W_t$$

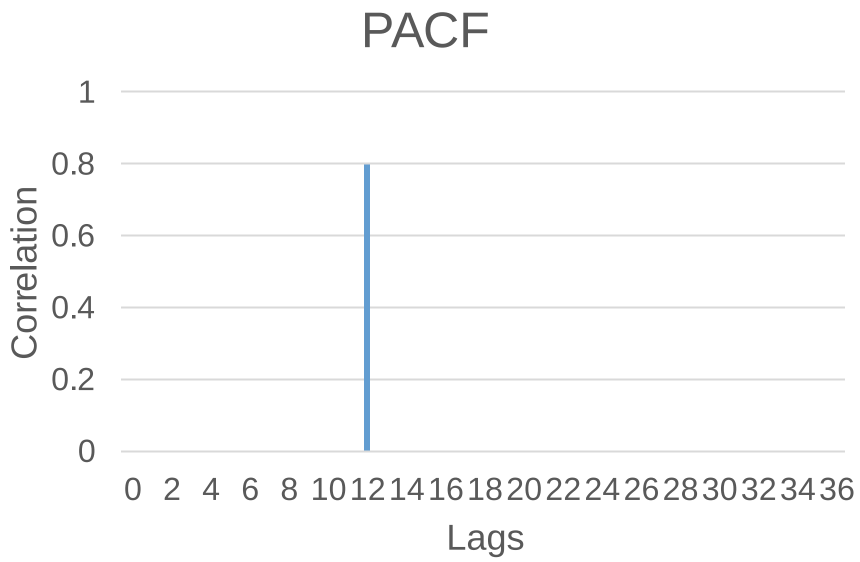
$$W_t = \omega + \phi_1 W_{t-1} + \phi_2 W_{t-12} + \phi_3 W_{t-24} + \theta_1 e_{t-1} + e_t$$

# Seasonal ARIMA

- Seasonal ARIMA models have the same structure and approach as typical ARIMA models with AR and MA patterns in the PACF and ACF.
- The pattern is just on the *seasonal* lag instead of the individual lags.

$$\text{ARIMA}(0,0,0)(1,0,0)_{12}$$

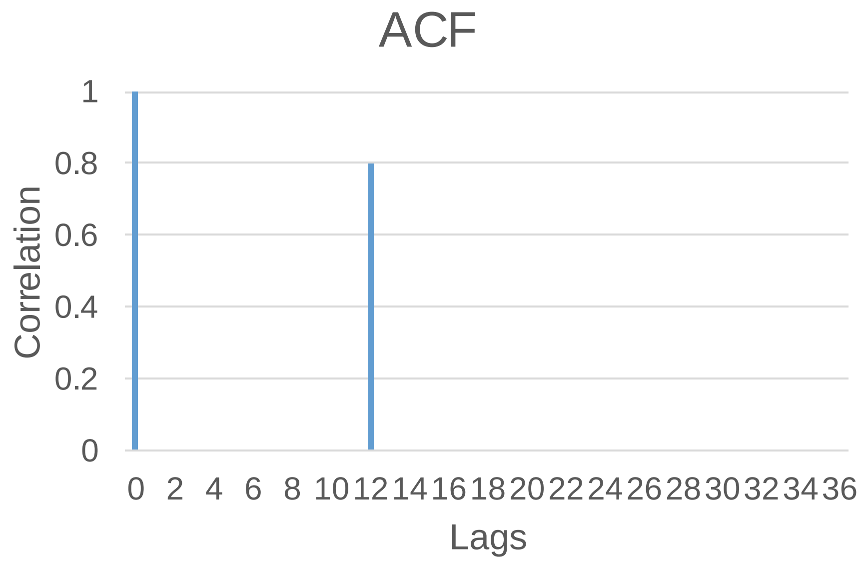


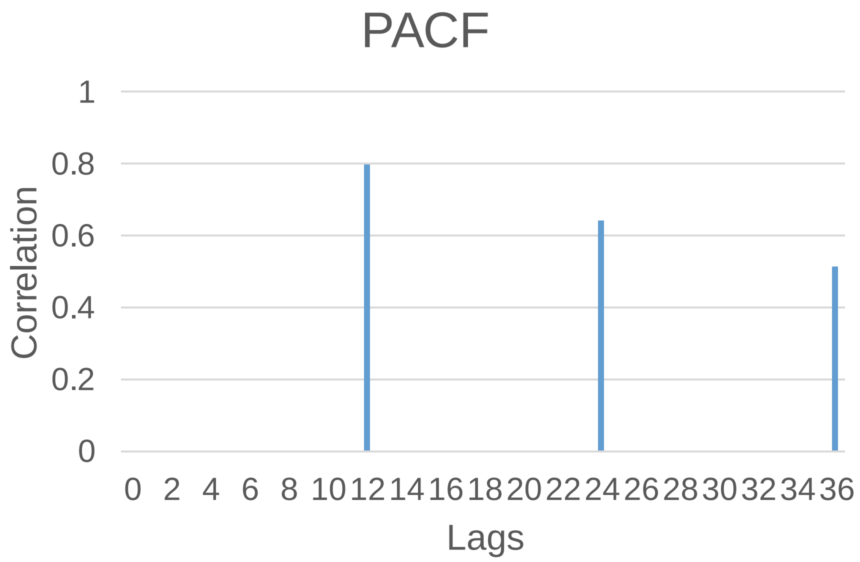



# Seasonal ARIMA

- Seasonal ARIMA models have the same structure and approach as typical ARIMA models with AR and MA patterns in the PACF and ACF.
- The pattern is just on the *seasonal* lag instead of the individual lags.

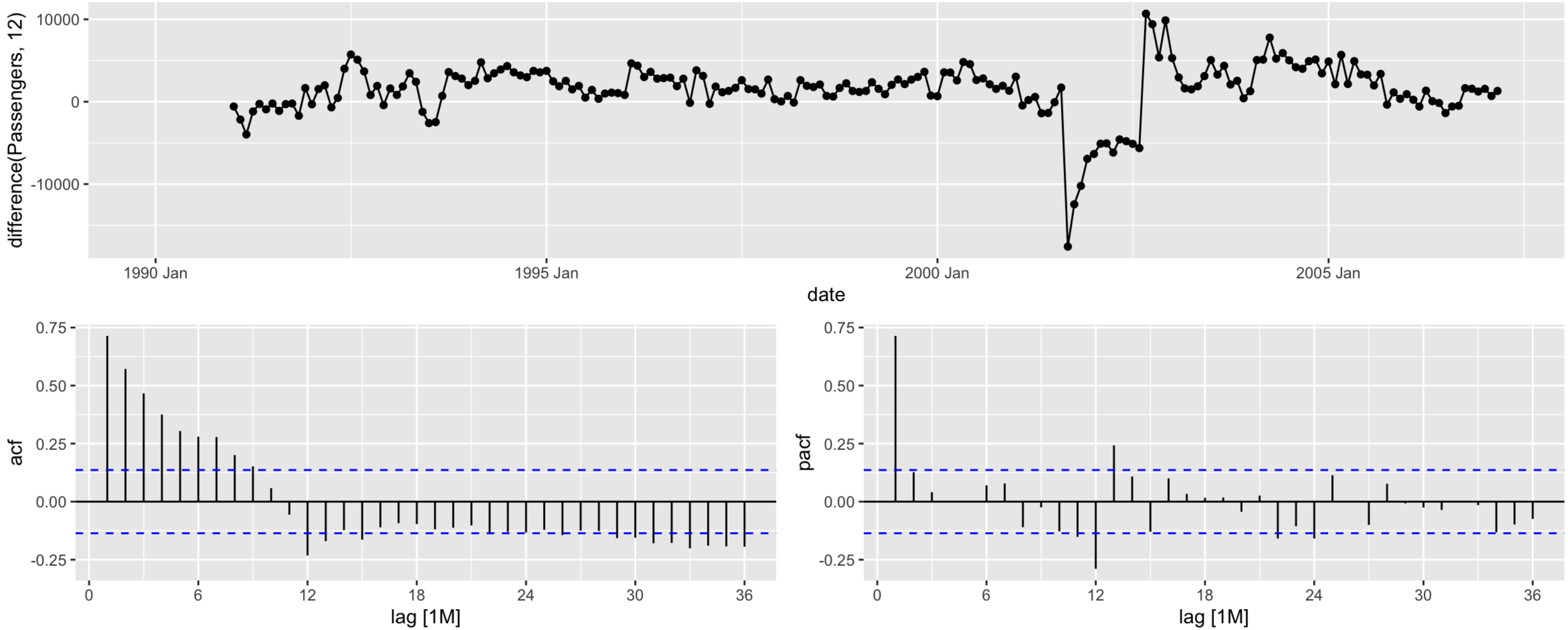
$$\text{ARIMA}(0,0,0)(0,0,1)_{12}$$



# Seasonal Differencing

```
train %>%
  gg_tsdisplay(difference(Passengers, 12), plot_type = 'partial', lag = 36)
```



# Seasonal ARIMA

```
model_SARIMA <- train %>%
  model(
    hand = ARIMA(Passengers ~ pdq(1,0,0) + PDQ(1,1,1)),
    auto = ARIMA(Passengers)
  )
```

# Seasonal ARIMA

```
model_SARIMA <- train %>%
  model(
    hand = ARIMA(Passengers ~ pdq(1,0,0) + PDQ(1,1,1)),
    auto = ARIMA(Passengers)
  )
```

Build specific ARIMA

Automatically select ARIMA

# Seasonal ARIMA – “by hand” model

```
model_SARIMA %>%
  select(hand) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "hand") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)

model_SARIMA %>%
  select(hand) %>%
  gg_tsresiduals(lag = 36)
```

# Seasonal ARIMA – “by hand” model

```
model_SARIMA %>%
  select(hand) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "hand") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)

model_SARIMA %>%
  select(hand) %>%
  gg_tsresiduals(lag = 36)
```

```
Series: Passengers
Model: ARIMA(1,0,0)(1,1,1)[12] w/ drift

Coefficients:
         ar1     sar1      sma1   constant
      0.7444   0.1721   -0.7755   319.7502
s.e.  0.0487   0.1040    0.0753    38.5404

sigma^2 estimated as 3745598:  log likelihood=-1754.8
AIC=3519.61   AICc=3519.93   BIC=3535.97
```

# Seasonal ARIMA – “by hand” model

```
model_SARIMA %>%
  select(hand) %>%
  report
```

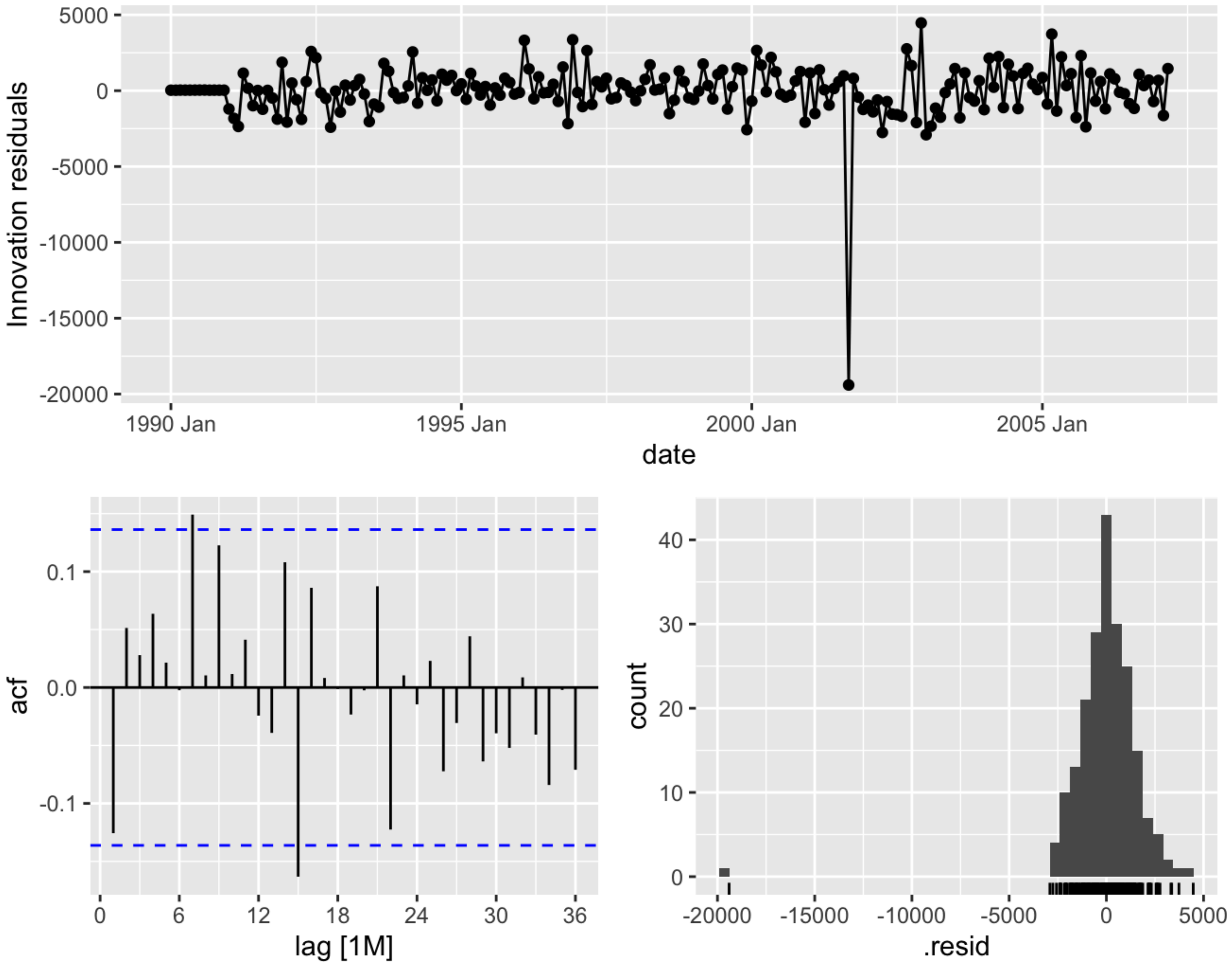
```
augment(model_SARIMA) %>%
  filter(.model == "hand") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)
```

```
# A tibble: 1 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 hand      37.4     0.275
```

```
model_SARIMA %>%
  select(hand) %>%
  gg_tsresiduals(lag = 36)
```

# Seasonal ARIMA – “by hand” model

```
model_SARIMA %>%
  select(hand) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "hand") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)

model_SARIMA %>%
  select(hand) %>%
  gg_tsresiduals(lag = 36)
```



# Seasonal ARIMA – “auto” model

```
model_SARIMA %>%
  select(auto) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "auto") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)

model_SARIMA %>%
  select(auto) %>%
  gg_tsresiduals(lag = 36)
```

# Seasonal ARIMA – "auto" model

```
model_SARIMA %>%
  select(auto) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "auto") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)

model_SARIMA %>%
  select(auto) %>%
  gg_tsresiduals(lag = 36)
```

```
Series: Passengers
Model: ARIMA(1,0,1)(0,1,1)[12] w/ drift

Coefficients:
         ar1      ma1     sma1  constant
      0.8801  -0.2962  -0.6785  179.8722
s.e.  0.0454   0.0950   0.0600   34.0147

sigma^2 estimated as 3639496:  log likelihood=-1751.67
AIC=3513.34   AICc=3513.66   BIC=3529.7
```

# Seasonal ARIMA – “auto” model

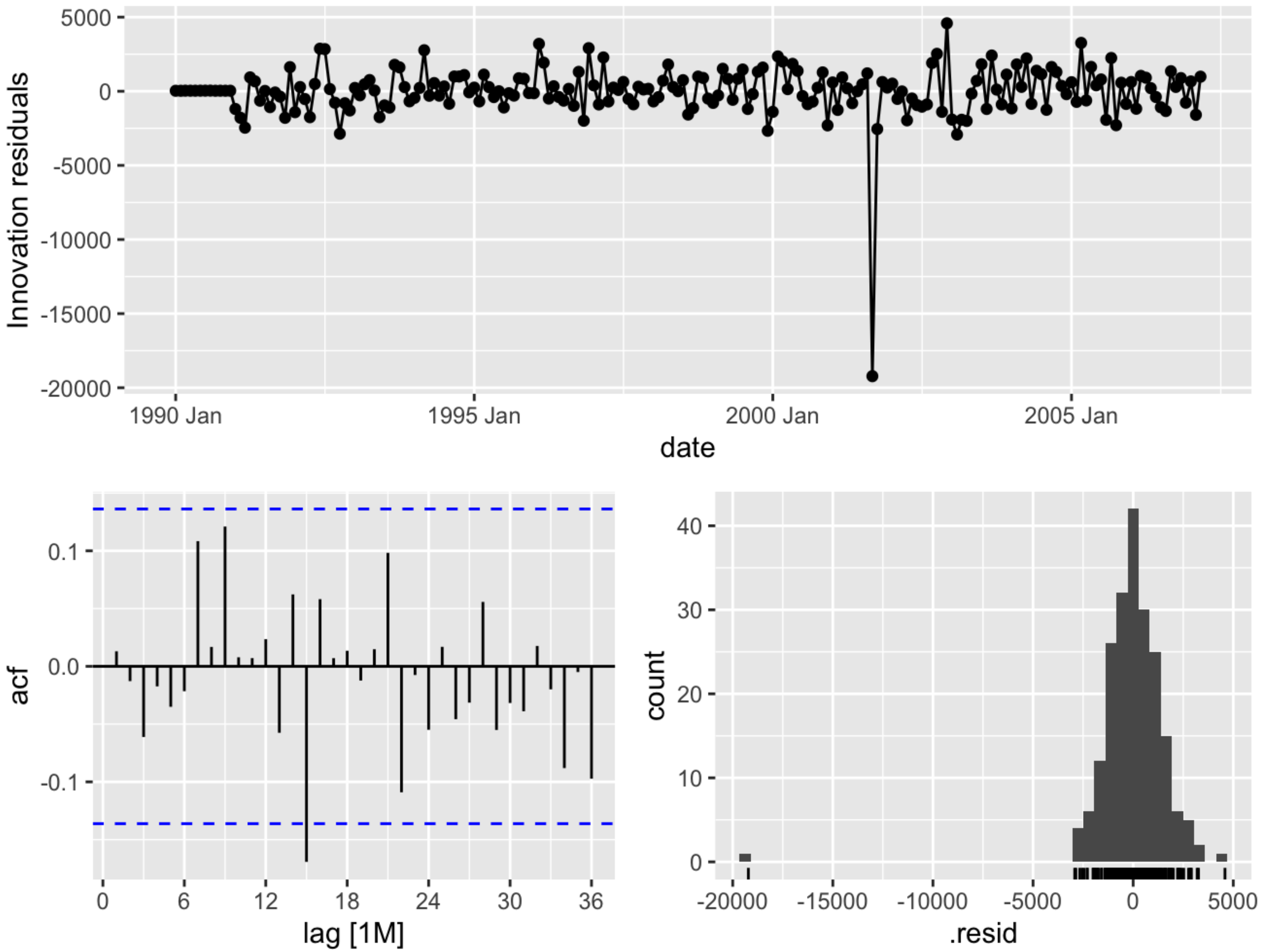
```
model_SARIMA %>%
  select(auto) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "auto") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)

model_SARIMA %>%
  select(auto) %>%
  gg_tsresiduals(lag = 36)
```

```
# A tibble: 1 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 auto      29.4     0.648
```

# Seasonal ARIMA – “auto” model

```
model_SARIMA %>%
  select(auto) %>%
  report

augment(model_SARIMA) %>%
  filter(.model == "auto") %>%
  features(.innov, ljung_box, lag = 36,
           dof = 3)
```

```
model_SARIMA %>%
  select(auto) %>%
  gg_tsresiduals(lag = 36)
```



# Multiple Differences

- Models can contain both unit roots and seasonal unit roots.
- After removing the seasonal unit root through differencing to get $W_t$, ordinary differences can be calculated.

$$W_t = Y_t - Y_{t-12}$$

$$W_t = W_{t-1} + e_t - \beta e_{t-1}$$

$$W_t - W_{t-1} = e_t - \beta e_{t-1}$$

$$(Y_t - Y_{t-12}) - (Y_{t-1} - Y_{t-13}) = e_t - \beta e_{t-1}$$

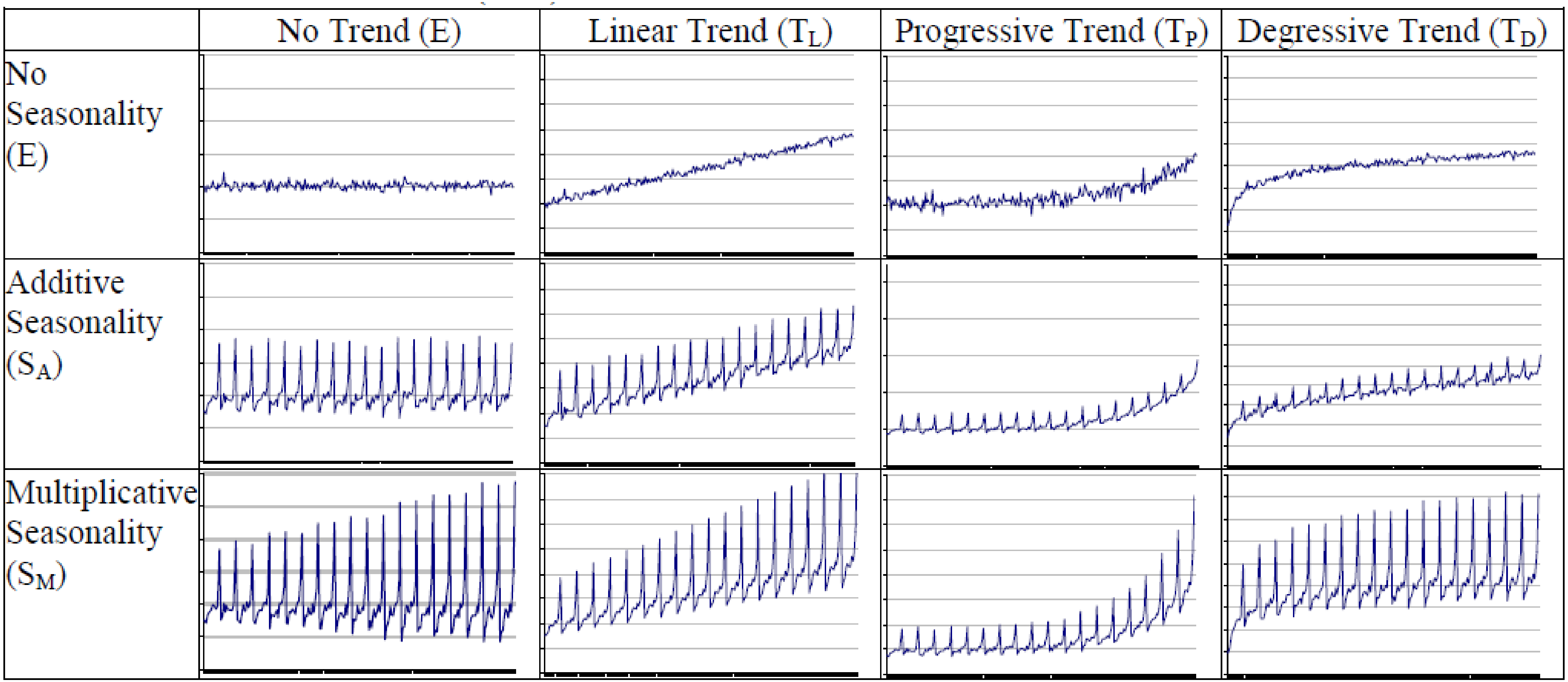
$$Y_t = Y_{t-1} + Y_{t-12} - Y_{t-13} + e_t - \beta e_{t-1}$$

# Limitations of Differencing

- Hard to evaluate stochastic effects for long and complex seasons.
- Most statistical tests for stochastic vs. deterministic can not handle past 12 or 24 periods in a season → need to use seasonal strength tests.
- Long/complex seasons → Best to just approach with deterministic solutions.

# MULTIPLICATIVE VS. ADDITIVE

# Multiplicative vs. Additive



# Backshift Operator – B

- The backshift operator is the mathematical operator to convert observations to their lags.
  - $\mathrm{B}(Y_t) = Y_{t-1}$

- This can be extended to any number of lags.
  - $\mathrm{B}^2(Y_t) = \mathrm{B}(Y_{t-1}) = Y_{t-2}$

# Structures to Seasons

Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

# Structures to Seasons

Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

$$Y_t - \alpha_1 \mathrm{B}(\mathrm{Y_t}) - \alpha_2 \mathrm{B}^{12}(Y_t) = e_t$$

Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

# Structures to Seasons

Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

$$Y_t - \alpha_1 \mathrm{B}(\mathrm{Y_t}) - \alpha_2 \mathrm{B}^{12}(Y_t) = e_t$$

$$Y_t - \alpha_1 Y_{t-1} - \alpha_2 Y_{t-12} = e_t$$

Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

# Structures to Seasons

Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

$$Y_t - \alpha_1 \mathrm{B}(\mathrm{Y_t}) - \alpha_2 \mathrm{B}^{12}(Y_t) = e_t$$

$$Y_t - \alpha_1 Y_{t-1} - \alpha_2 Y_{t-12} = e_t$$

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-12} + e_t$$

Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

# Structures to Seasons

Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12}) Y_t = e_t$$

Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12}) Y_t = e_t$$

$$\Downarrow$$

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12} + \alpha_1 \alpha_2 \mathrm{B}^{13}) Y_t = e_t$$

# Structures to Seasons

Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12} - \boxed{\alpha_3} \mathrm{B}^{13})Y_t = e_t$$

Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12})Y_t = e_t$$

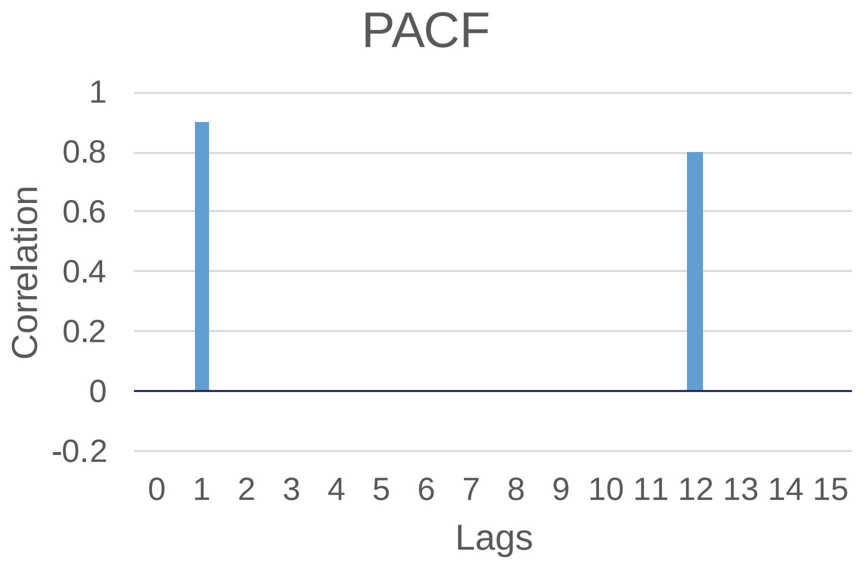
$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12} + \boxed{\alpha_1 \alpha_2} \mathrm{B}^{13})Y_t = e_t$$

?

# Structures to Seasons

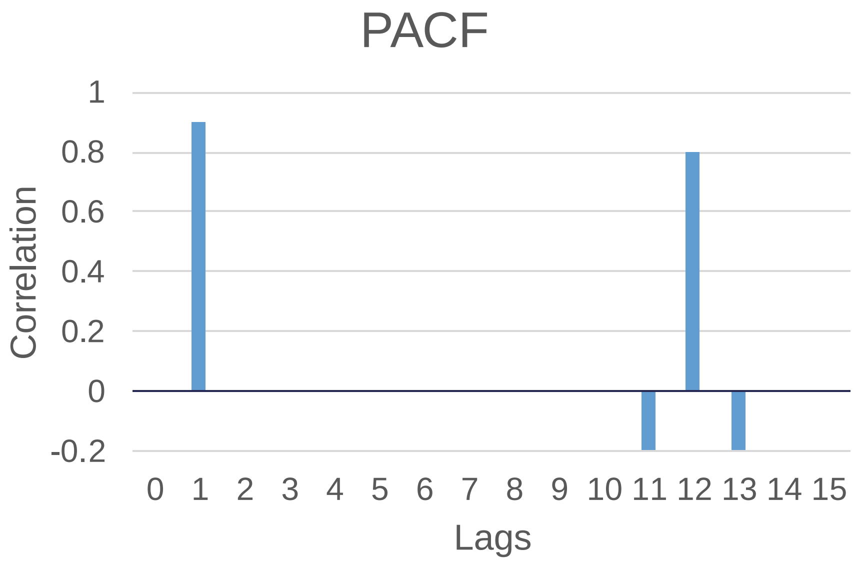
## Additive

$$(1 - \alpha_1 \mathrm{B} - \alpha_2 \mathrm{B}^{12}) Y_t = e_t$$



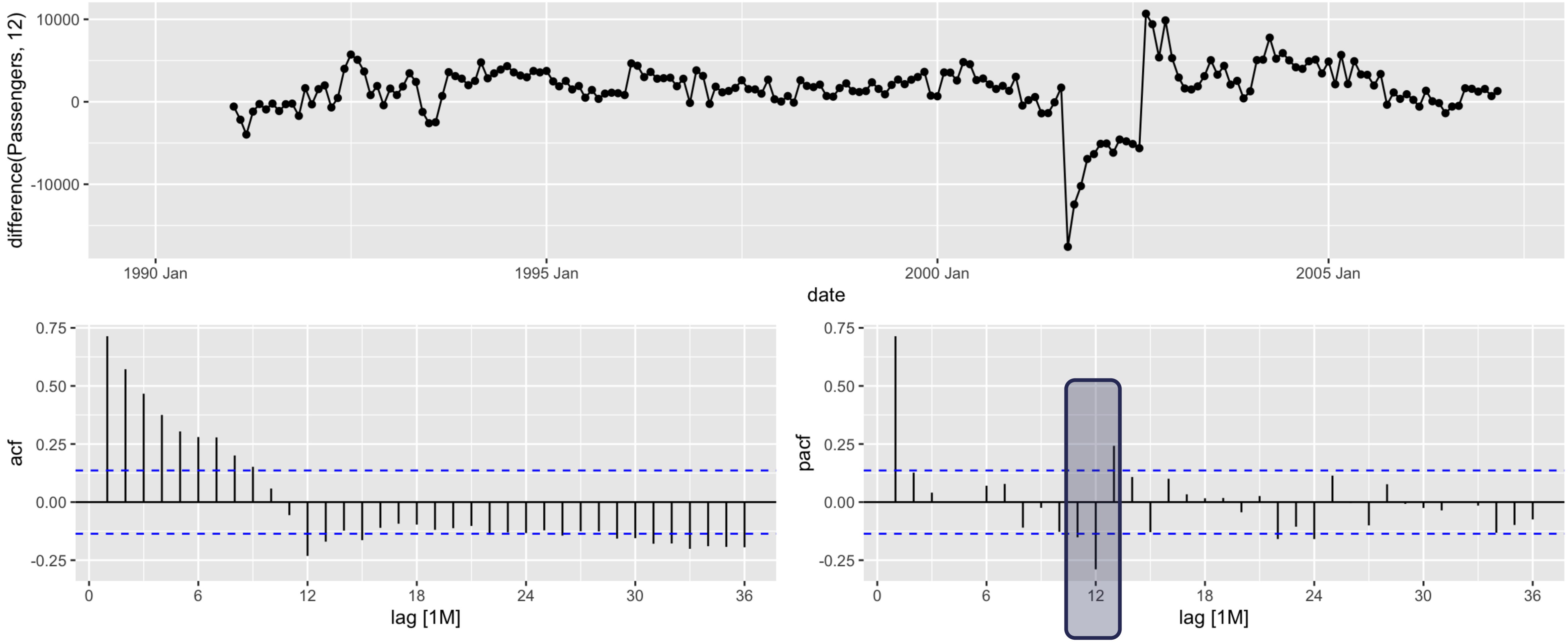

## Multiplicative

$$(1 - \alpha_1 \mathrm{B})(1 - \alpha_2 \mathrm{B}^{12}) Y_t = e_t$$



# Seasonal Differencing

```
train %>%
  gg_tsdisplay(difference(Passengers, 12), plot_type = 'partial', lag = 36)
```



# Structures to Seasons

## Additive

LOTS OF ANNOYING CODING IN R!

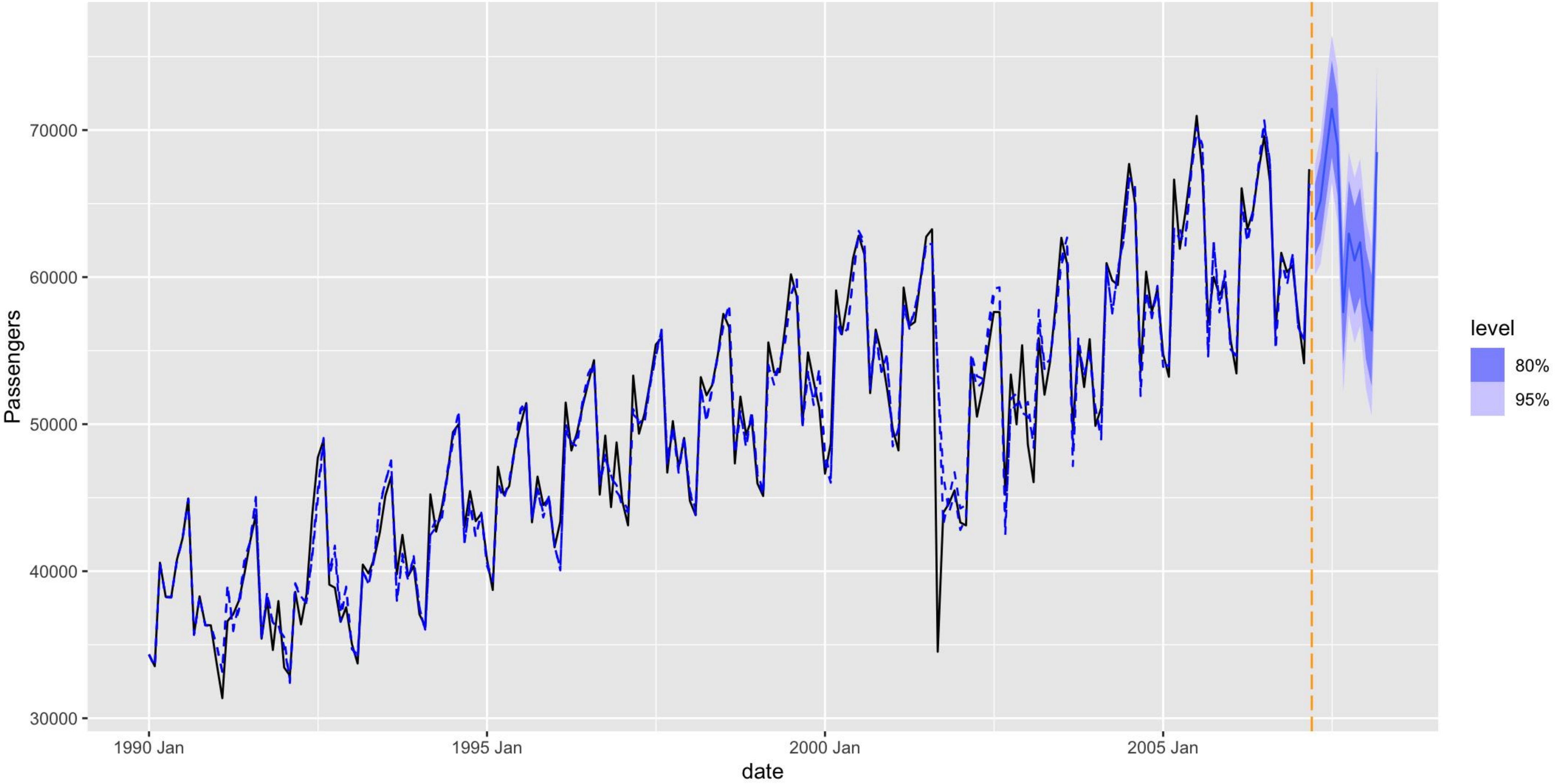
## Multiplicative – DEFAULT

```
model_SARIMA <- train %>%
  model(
    hand = ARIMA(Passengers ~ pdq(1,0,0)
                 + PDQ(1,1,1)),
    auto = ARIMA(Passengers)
  )
```

# Seasonal ARIMA Models

```
model_SARIMA_for <- forecast(model_SARIMA, h = 12)

fabletools::accuracy(model_SARIMA_for, test)
```

# Seasonal ARIMA Models



# Model Evaluation on Test Data

| Model | MAE | MAPE |
|---|---|---|
| HW Exponential Smoothing | 1100.02 | 1.71% |
| Seasonal ARIMA – AUTO | 1229.22 | 1.89% |
| Seasonal ARIMA – Dr L | 1161.71 | 1.78% |

# Model Evaluation on Test Data

| Model | MAE | MAPE |
|---|---|---|
| HW Exponential Smoothing | 1100.02 | 1.71% |
| Seasonal ARIMA | 1161.71 | 1.78% |

# ?