ARIMA models: Addressing seasonality

EC 361–001

Prof. Santetti
Spring 2024

Materials

Required readings:

- `Hyndman & Athanasopoulos, ch. 9`
    - sections 9.9—9.10.
    - reading 9.10 is *optional*.

Motivation

Motivation

By now, we know that time series that show prominent **trends** will *not* be stationary.

In addition, when **seasonality** is present, the time series **will not be stationary**.

There are specific **ARIMA** models that deal with seasonal series, and this is what this lecture is about.

Seasonal ARIMA models

Seasonal ARIMA models

A **seasonal ARIMA** model is formed by including **additional seasonal terms** in the ARIMA models we have seen so far.

A seasonal ARIMA model can be expressed as follows:

$$\text{ARIMA} \quad \underbrace{(p, d, q)}_{\text{Non-seasonal part}} \quad \underbrace{(P, D, Q)_m}_{\text{Seasonal part}}$$

Notice that we will adopt **upper-case** letters *(P, D, Q)* to express the ARIMA terms of the seasonal component.

Seasonal ARIMA models

For example, an **ARIMA(1, 1, 1) $(2, 1, 1)_4$** denotes a seasonal ARIMA model for *quarterly* data ($m = 4$), where to achieve stationarity we need:

- to take first differences ($d = 1$);
- to take an additional seasonal difference ($D = 1$).

In addition, the **seasonal** part has *two* autoregressive and *one* moving-average component, while the **non-seasonal** part has one *AR* and one *MA* component.

In terms of **order selection**, we still make use of **ACF** and **PACF** plots for help.

Seasonal ARIMA models

The seasonal part of an *ARIMA* model will be seen in the **seasonal lags** of the PACF and ACF plots.

As an example, consider an ***ARIMA(0, 0, 0) (0, 0, 1)$_{12}$*** model.

It will show the following features:

- A *spike* at lag 12 in the ACF but no other significant spikes;
- exponential *decay* in the seasonal lags of the PACF (i.e., at lags 12, 24, 36, ...).

In a similar way, for an ***ARIMA(0, 0, 0) (1, 0, 0)$_{12}$*** model, we will see:

- Exponential *decay* in the seasonal lags of the ACF;
- A single significant *spike* at lag 12 in the PACF.

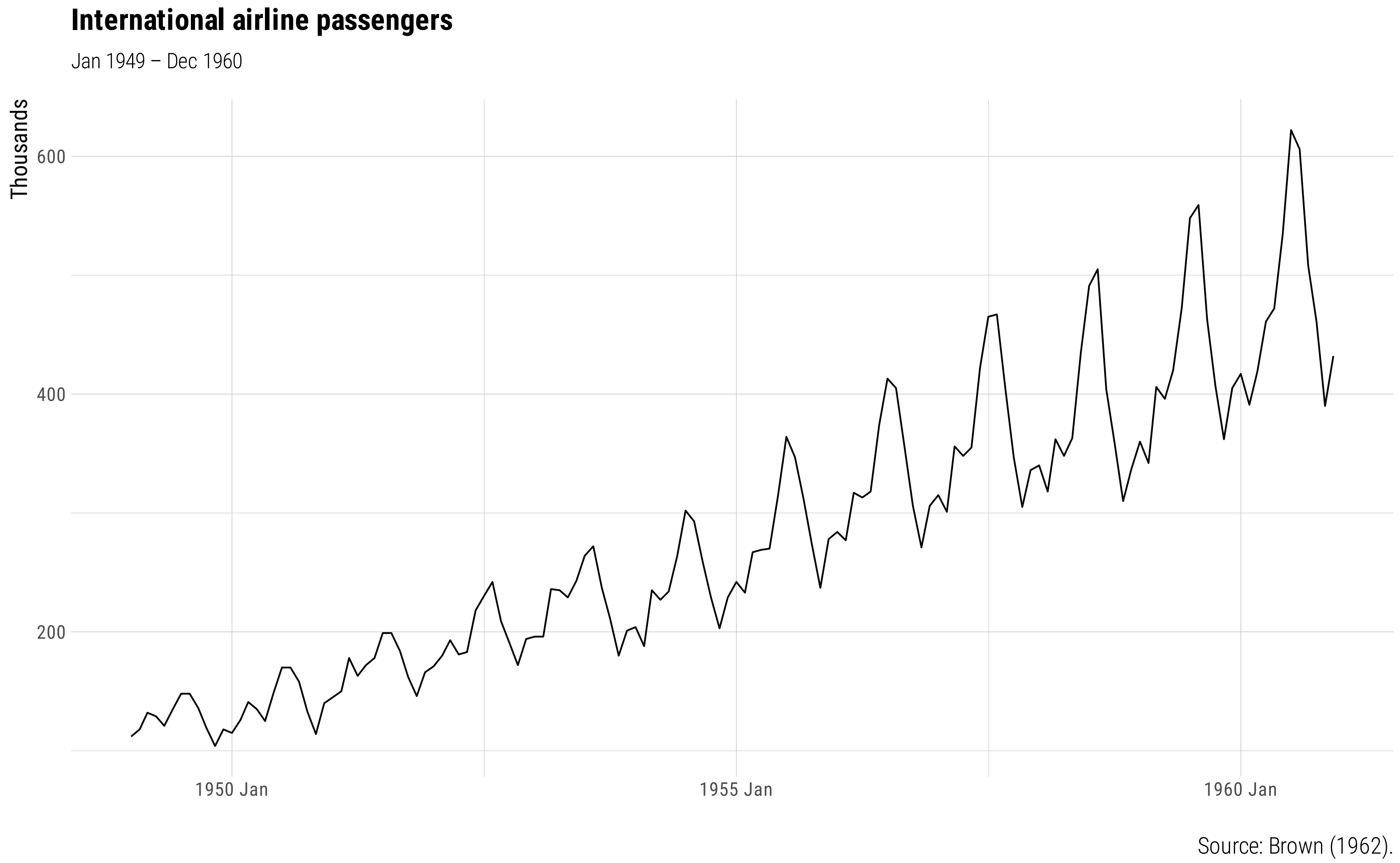
Thus, when deciding on the P and Q values of a seasonal ARIMA model, we must restrict our attention to the **seasonal lags**.

Seasonal ARIMA models

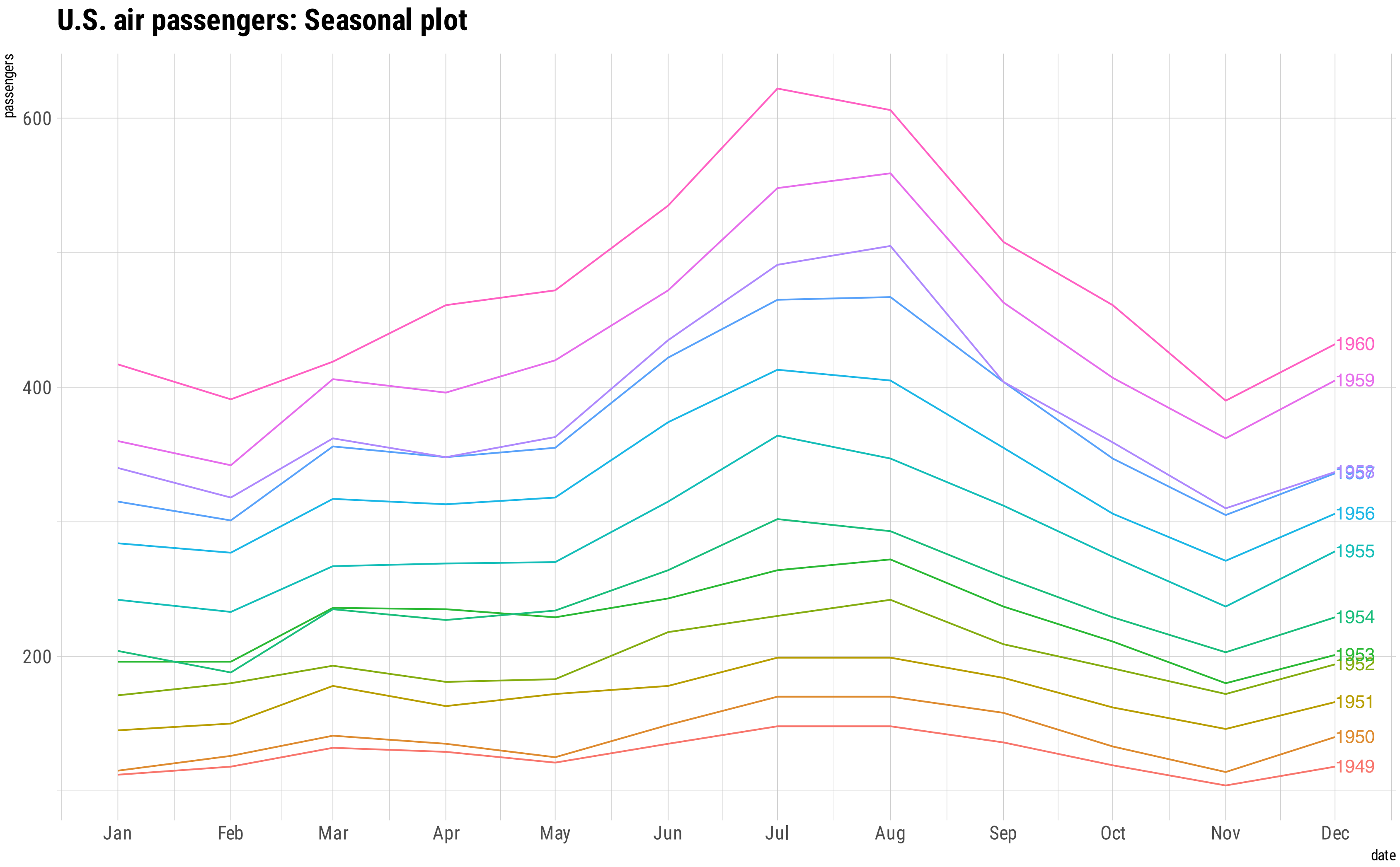
The **Hyndman-Khandakar algorithm** looks and works basically in the same way as seen in the previous lecture.

It will only differ as additional parameters (namely P, D, and Q) need to be included in the mix.

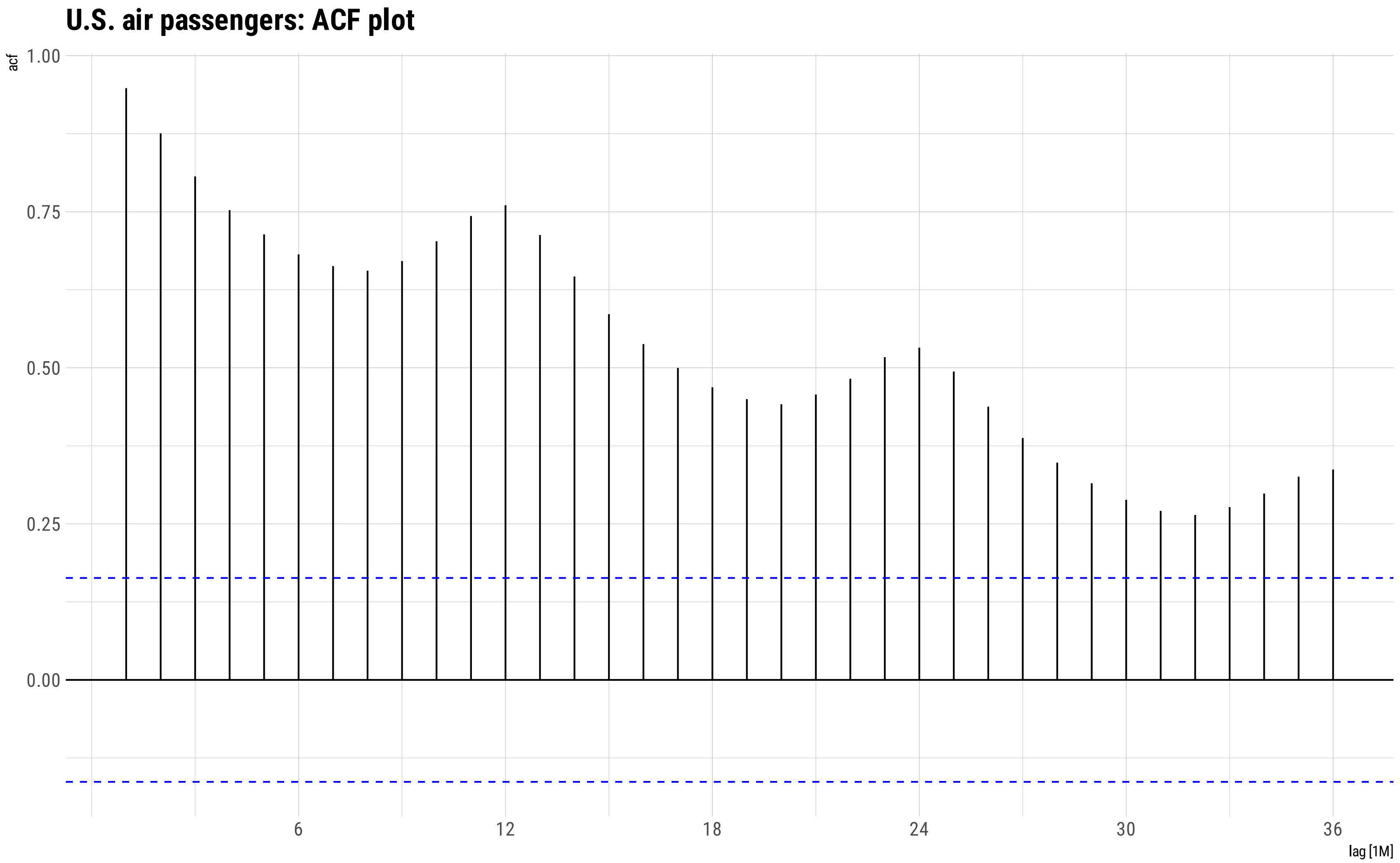
Seasonal ARIMA models



Seasonal ARIMA models



Seasonal ARIMA models



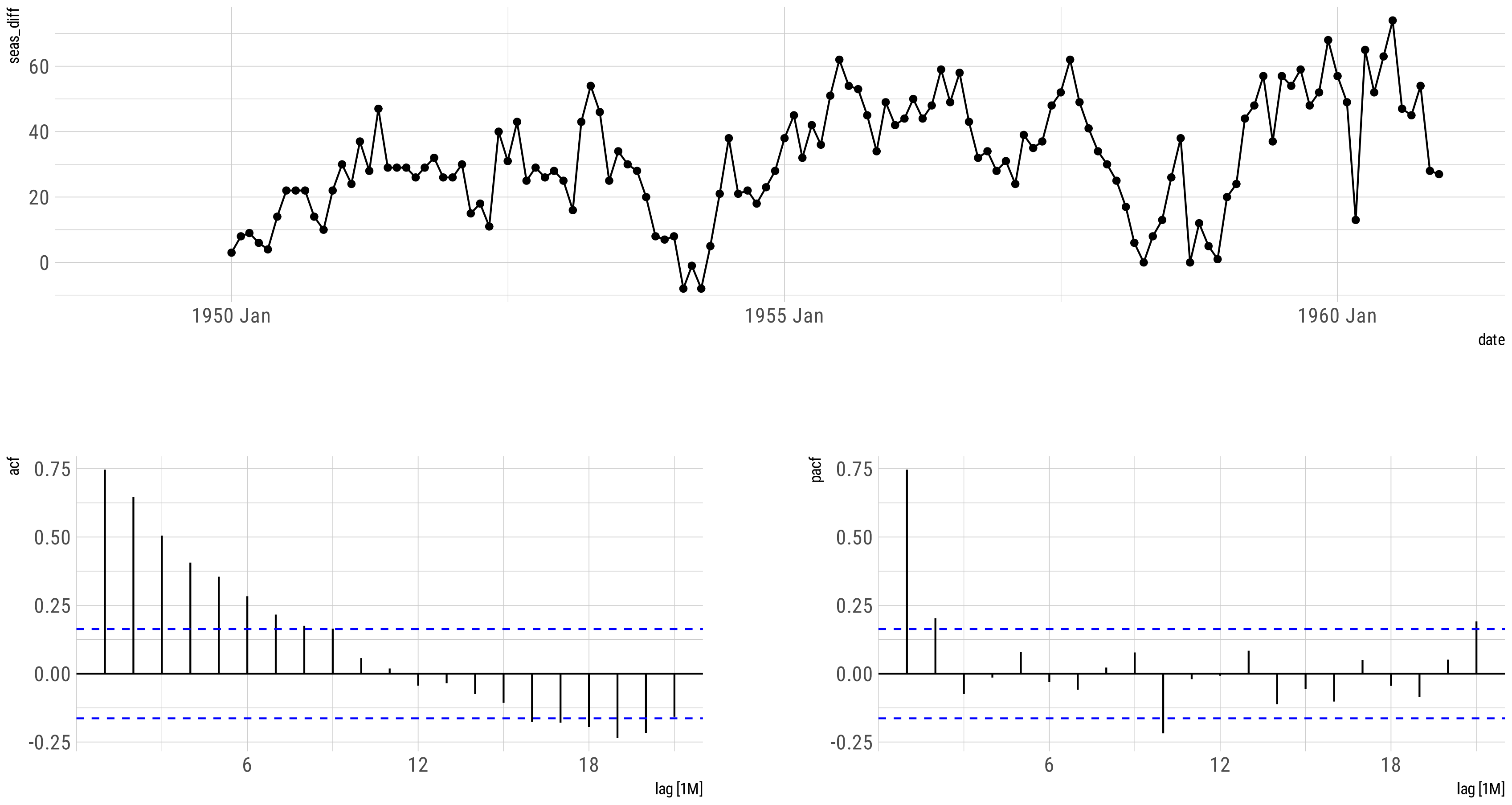
Seasonal ARIMA models

Is the series stationary?

```
air_ts ▷
  features(passengers, unitroot_kpss)
```

```
#> # A tibble: 1 × 2
#>   kpss_stat kpss_pvalue
#>       <dbl>       <dbl>
#> 1      2.74        0.01
```

```r
air_ts |>
  mutate(seas_diff = difference(passengers, lag = 12)) |> # taking seasonal differencing.
  gg_tsdisplay(seas_diff, plot_type = "partial")
```



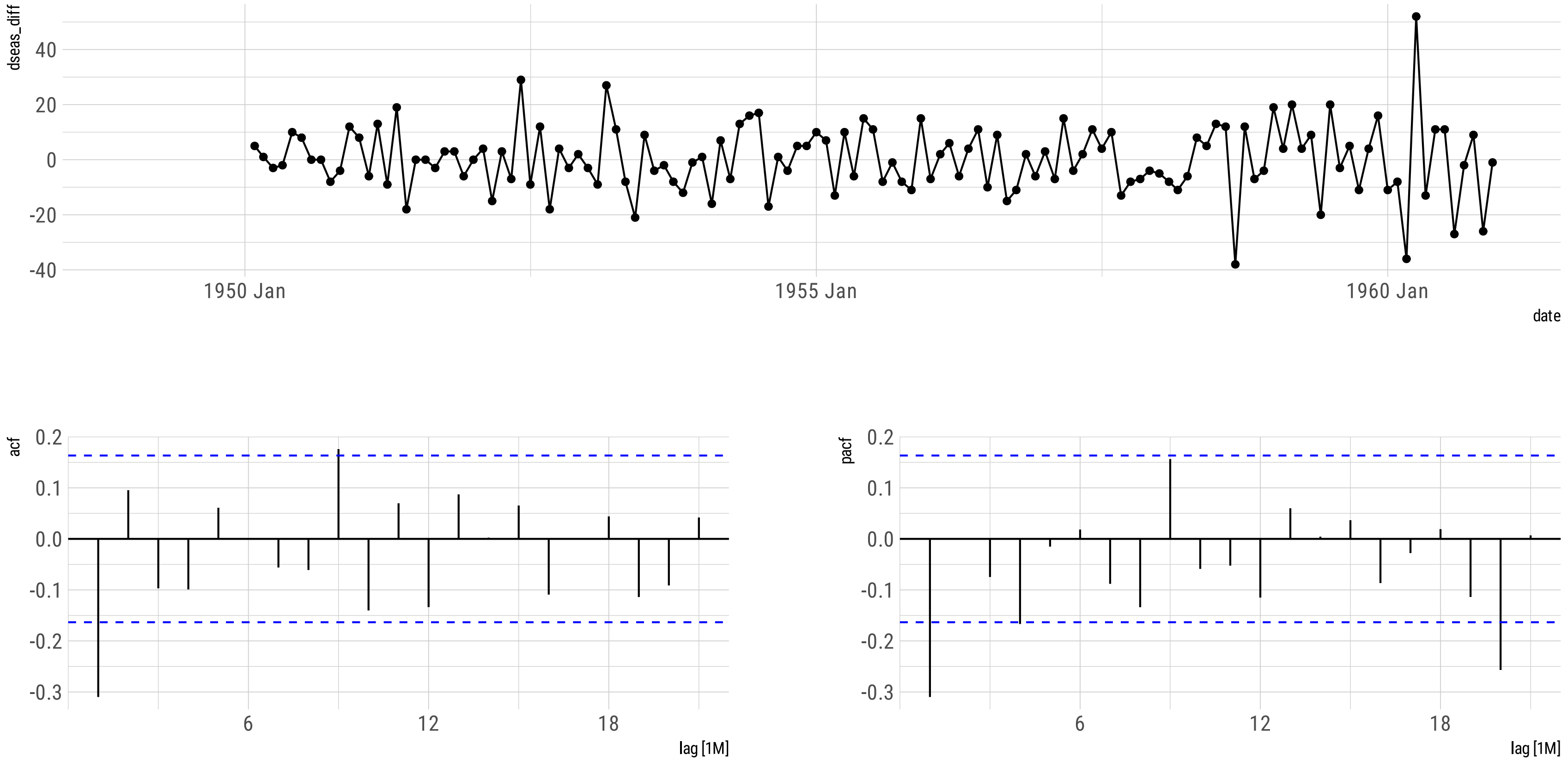
Seasonal ARIMA models

Is the **seasonally differenced** series stationary?

```
air_ts ▷
  mutate(seas_diff = difference(passengers, lag = 12)) ▷
  features(seas_diff, unitroot_kpss)
```

```
#> # A tibble: 1 × 2
#>   kpss_stat kpss_pvalue
#>       <dbl>       <dbl>
#> 1     0.666      0.0167
```

```r
air_ts |>
  mutate(seas_diff = difference(passengers, lag = 12),
         dseas_diff = difference(seas_diff, lag = 1)) |>  #taking 1st difference of seasonal difference.
  gg_tsdisplay(dseas_diff, plot_type = "partial")
```



Seasonal ARIMA models

Have we achieved stationarity?

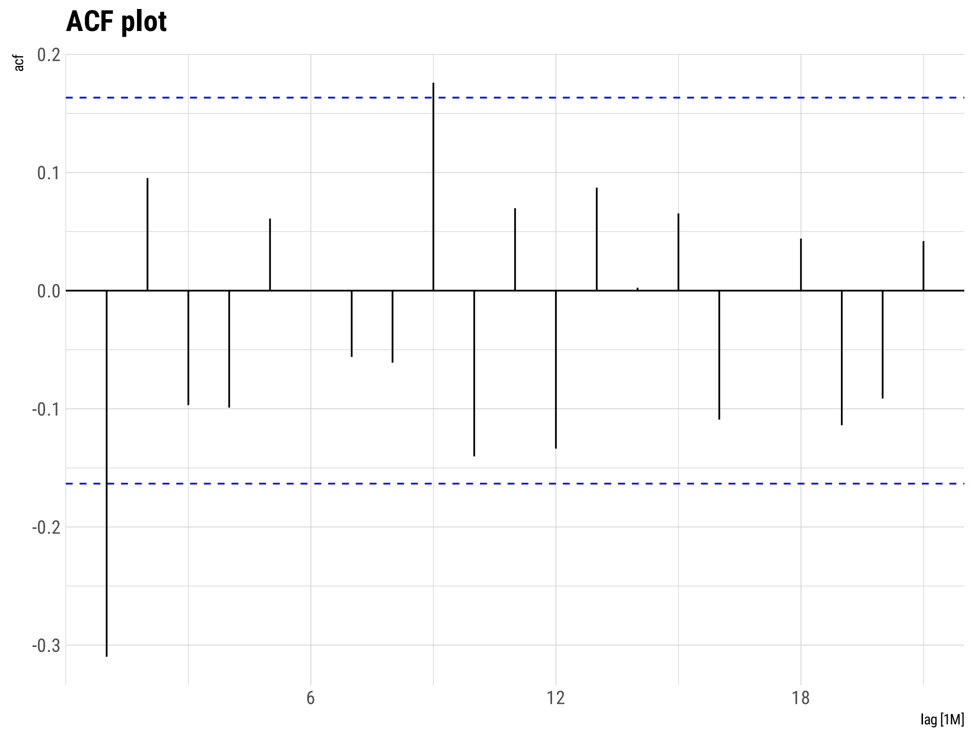
```
air_ts ▷
  mutate(seas_diff = difference(passengers, lag = 12),
         dseas_diff = difference(seas_diff, lag = 1)) ▷
  features(dseas_diff, unitroot_kpss)
```

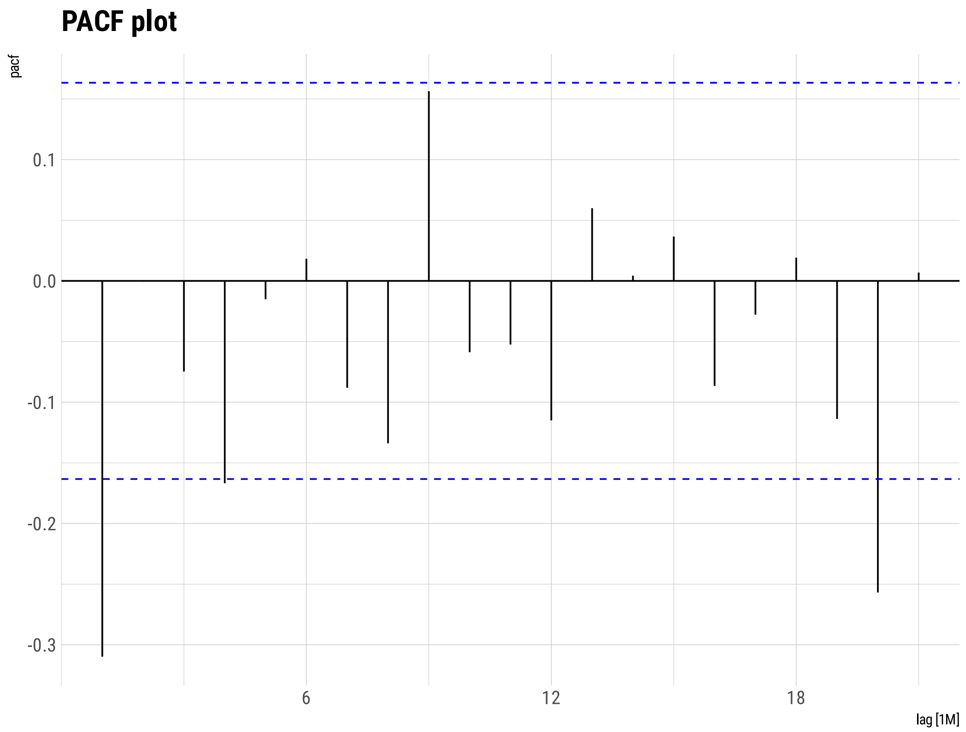
```
#> # A tibble: 1 × 2
#>   kpss_stat kpss_pvalue
#>       <dbl>       <dbl>
#> 1    0.0543         0.1
```

Let us use the **ACF** and **PACF** plots to decide on the appropriate AR and MA orders.

```r
air_ts |>
  ACF(dseas_diff) |>
  autoplot() +
  labs(title = "ACF plot")
```

```r
air_ts |>
  PACF(dseas_diff) |>
  autoplot() +
  labs(title = "PACF plot")
```





Seasonal ARIMA models

Estimating a few models:

```
air_arima_fit <- air_ts |>
  model(arima110_010 = ARIMA(passengers ~ pdq(1, 1, 0) + PDQ(0, 1, 0)),
        arima011_010 = ARIMA(passengers ~ pdq(0, 1, 1) + PDQ(0, 1, 0)),
        arima310_010 = ARIMA(passengers ~ pdq(3, 1, 0) + PDQ(0, 1, 0)),
        arima_auto = ARIMA(passengers))

air_arima_fit
```

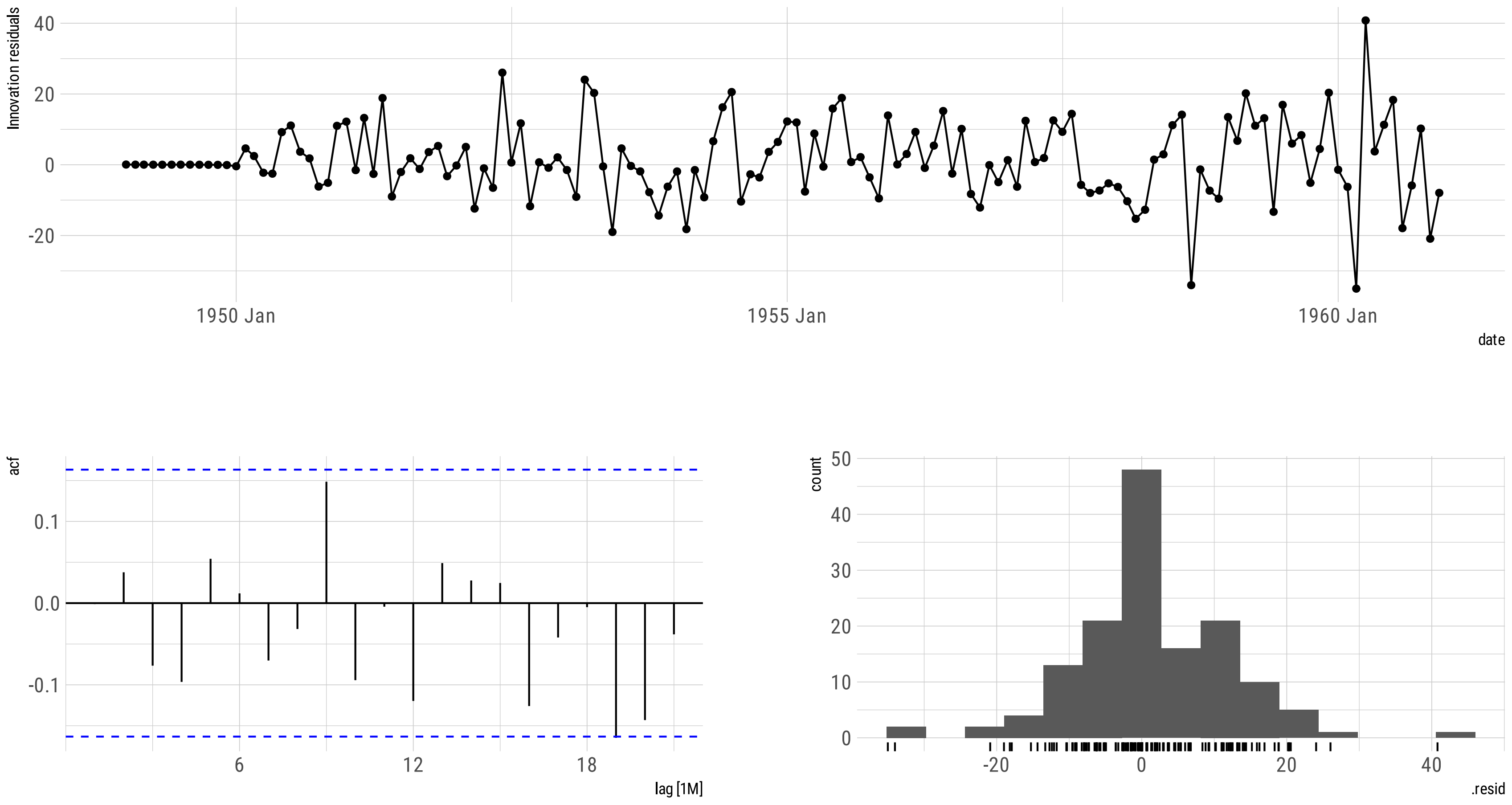
```
#> # A mable: 1 x 4
#>               arima110_010              arima011_010              arima310_010                arima_auto
#>                    <model>                   <model>                   <model>                   <model>
#> 1 <ARIMA(1,1,0)(0,1,0)[12]> <ARIMA(0,1,1)(0,1,0)[12]> <ARIMA(3,1,0)(0,1,0)[12]> <ARIMA(2,1,1)(0,1,0)[12]>
```

Seasonal ARIMA models

```
air_arima_fit ▷
  glance() ▷
  arrange(AICc) ▷
  select(.model, AIC, AICc)
```

```
#> # A tibble: 4 × 3
#>   .model         AIC  AICc
#>   <chr>        <dbl> <dbl>
#> 1 arima_auto   1018. 1018.
#> 2 arima110_010 1020. 1020.
#> 3 arima011_010 1021. 1021.
#> 4 arima310_010 1024. 1024.
```

```
air_arima_fit ▷
  select(arima_auto) ▷
  gg_tsresiduals()
```



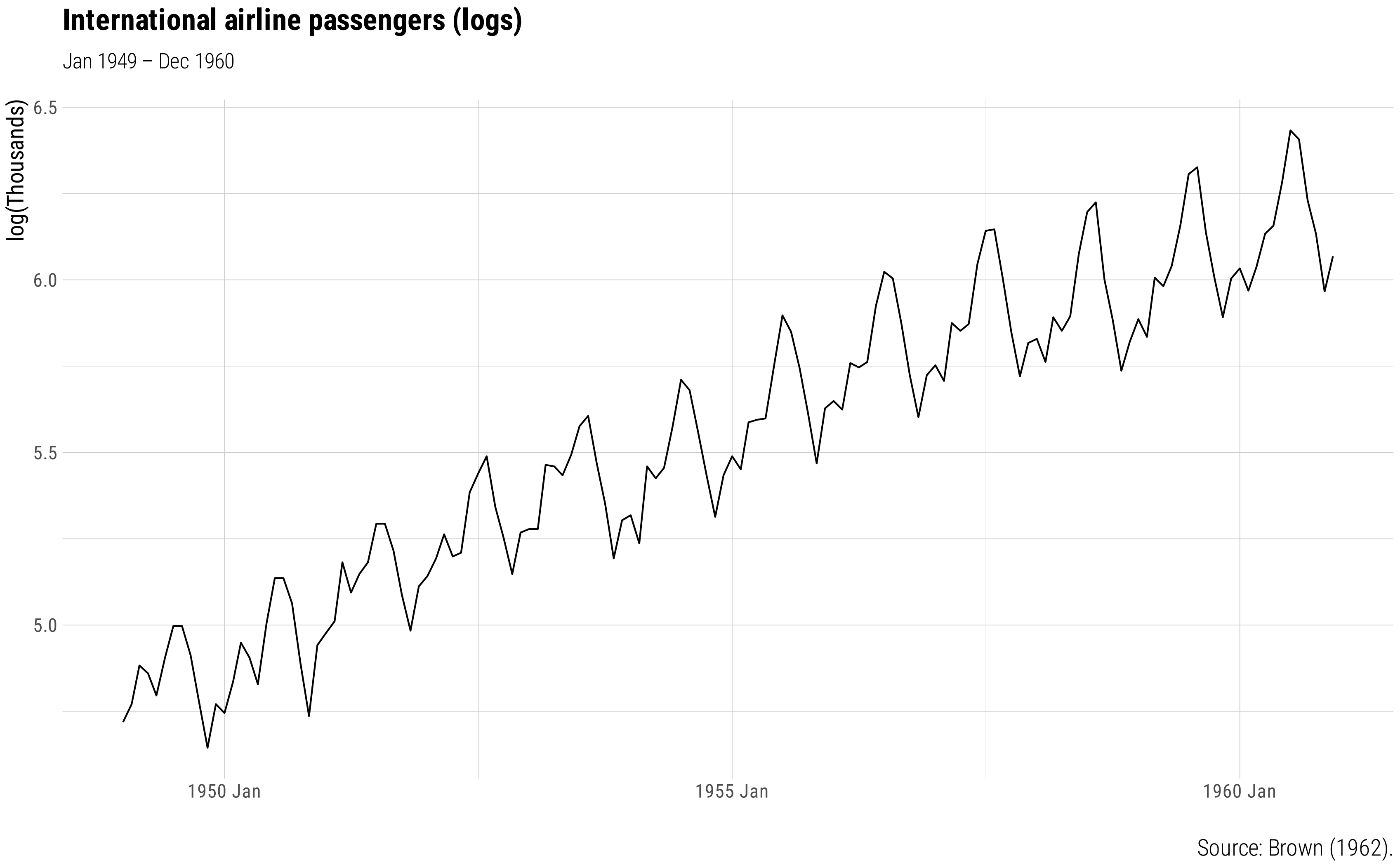
Seasonal ARIMA models

Are residuals **white noise**?

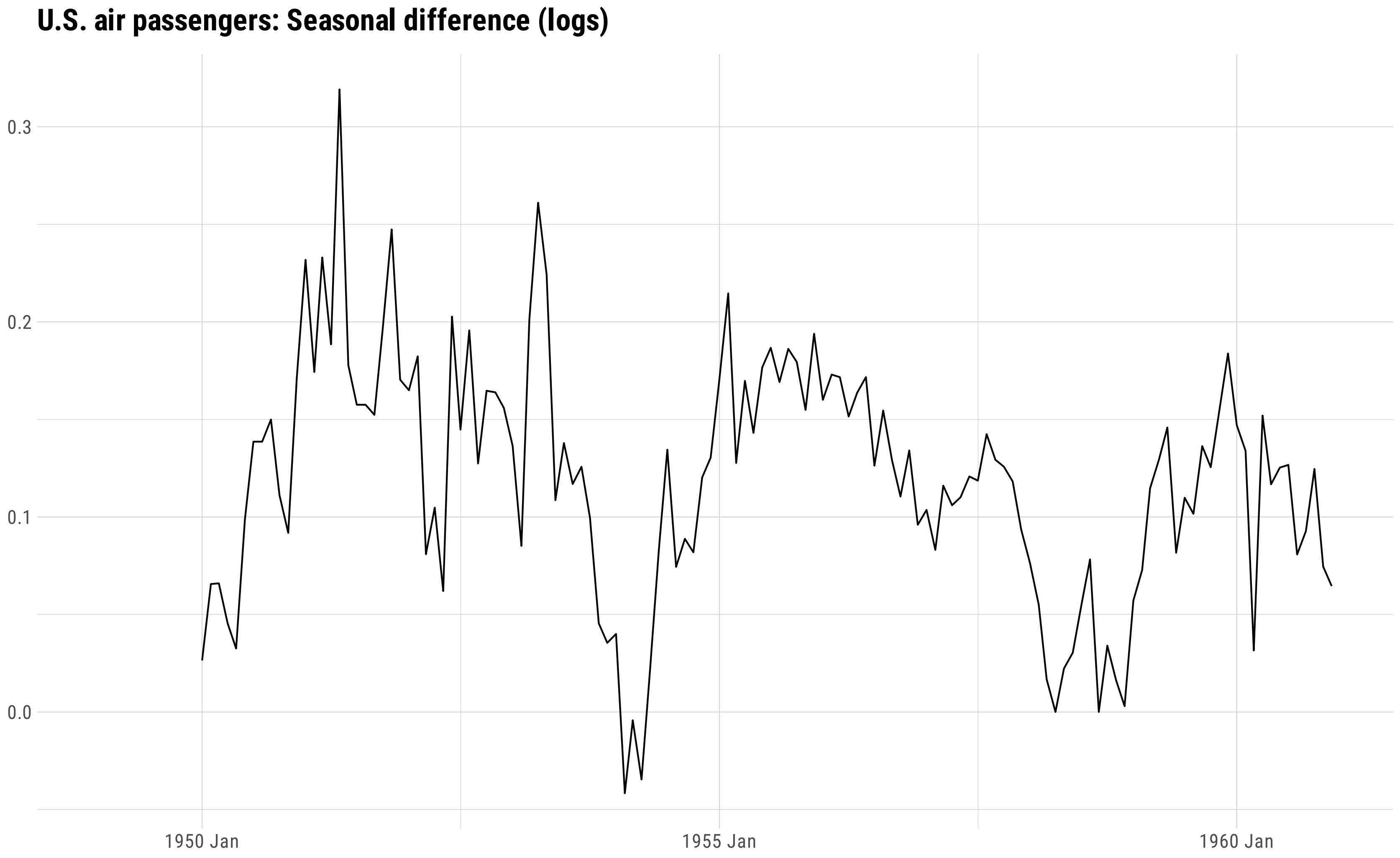
```r
air_arima_fit |>
  augment() |>
  filter(.model == "arima_auto") |>
  features(.innov, ljung_box, lag = 2 * 12, dof = 3)
```

```
#> # A tibble: 1 × 3
#>   .model     lb_stat lb_pvalue
#>   <chr>        <dbl>     <dbl>
#> 1 arima_auto    37.8    0.0137
```

Seasonal ARIMA models



Seasonal ARIMA models



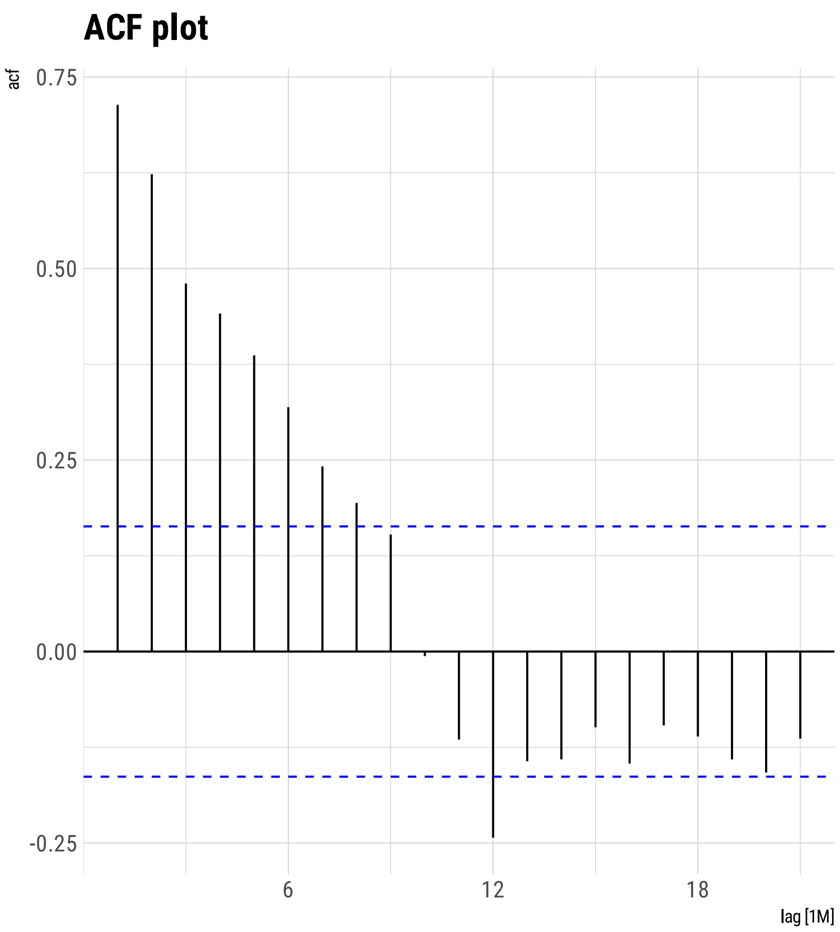
Seasonal ARIMA models

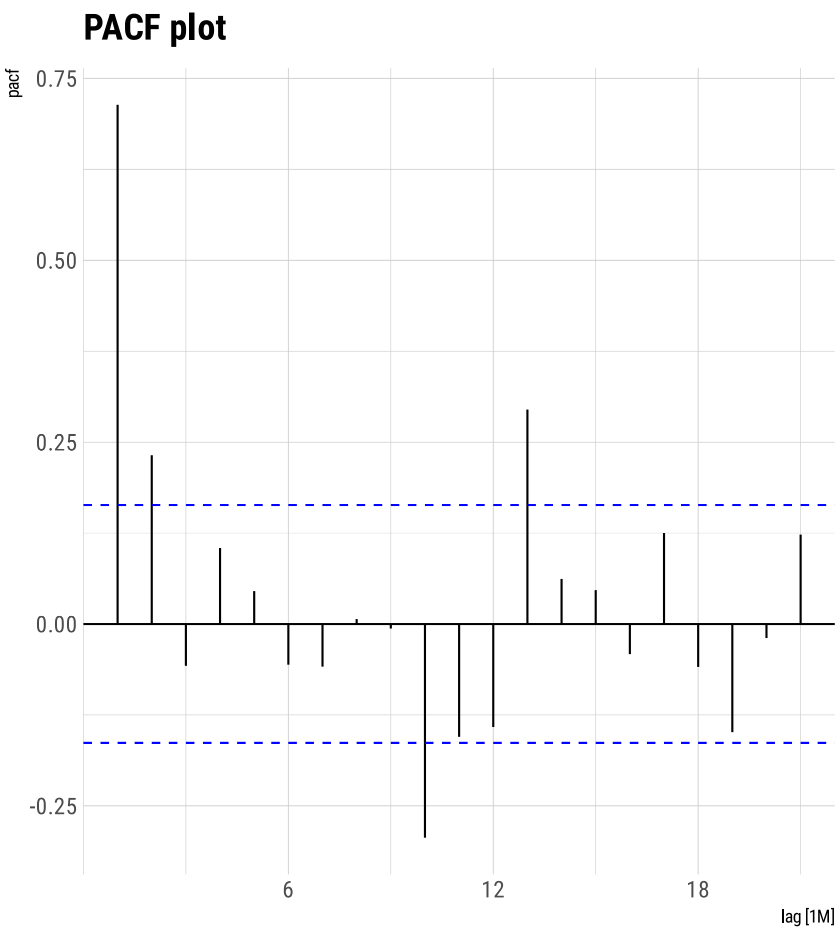
```
air_ts <- air_ts |>
  mutate(log_seas_diff = difference(log(passengers), lag = 12))

air_ts |>
  features(log_seas_diff, unitroot_kpss)
```

```
#> # A tibble: 1 × 2
#>   kpss_stat kpss_pvalue
#>       <dbl>       <dbl>
#> 1     0.368      0.0909
```

Seasonal ARIMA models





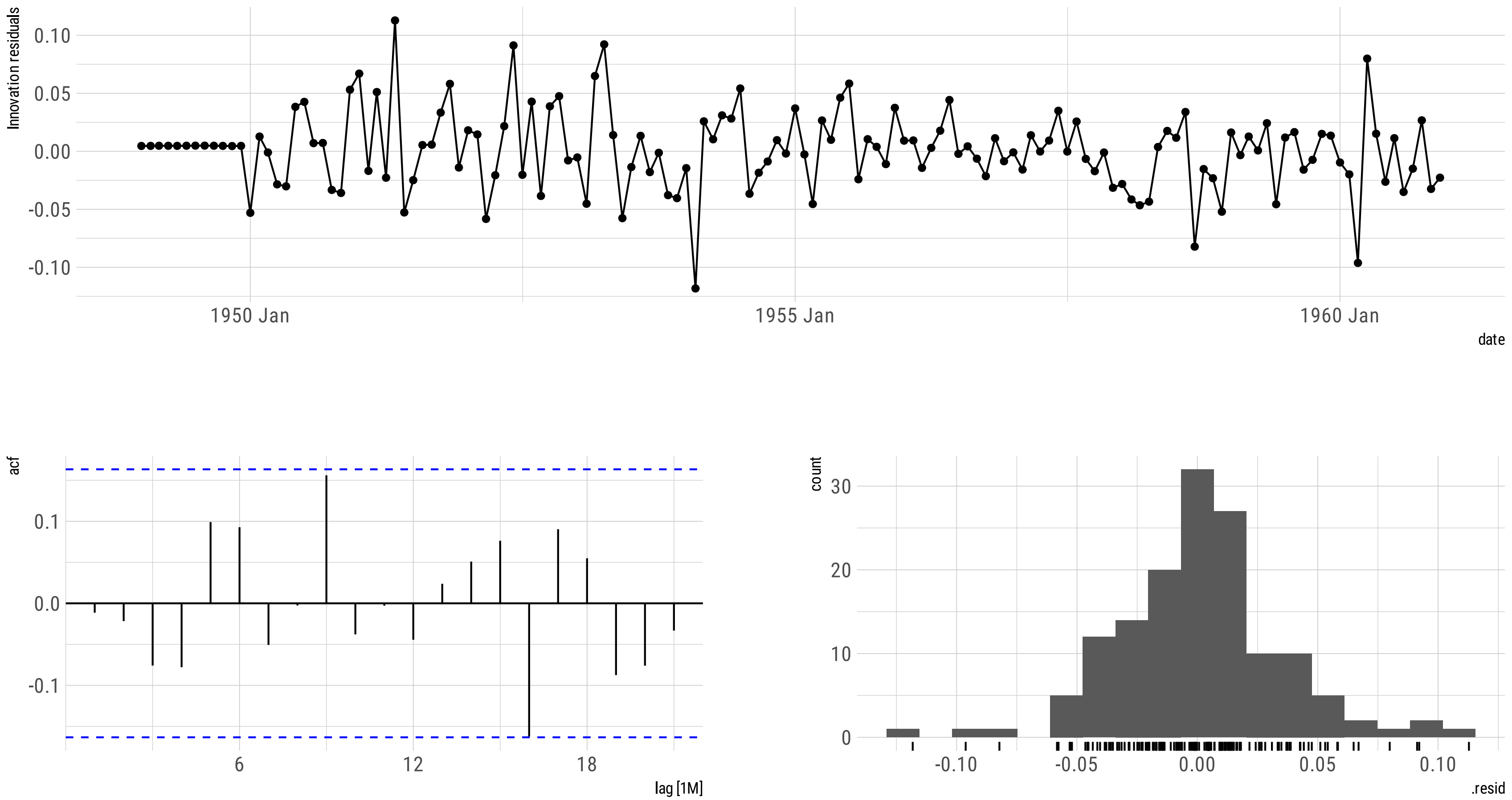
Seasonal ARIMA models

```
air_log_fit <- air_ts |>
  model(arima200_011 = ARIMA(log(passengers) ~ 1 + pdq(2, 0, 0) + PDQ(0, 1, 1)),
        arima202_011 = ARIMA(log(passengers) ~ 1 + pdq(2, 0, 2) + PDQ(0, 1, 1)),
        arima_auto = ARIMA(log(passengers)))

air_log_fit |>
  glance() |>
  arrange(AICc) |>
  select(.model, AIC, AICc)
```

```
#> # A tibble: 3 × 3
#>   .model         AIC  AICc
#>   <chr>        <dbl> <dbl>
#> 1 arima200_011 -489. -489.
#> 2 arima_auto   -489. -489.
#> 3 arima202_011 -486. -485.
```

```
air_log_fit ▷
  select(arima_auto) ▷
  gg_tsresiduals()
```



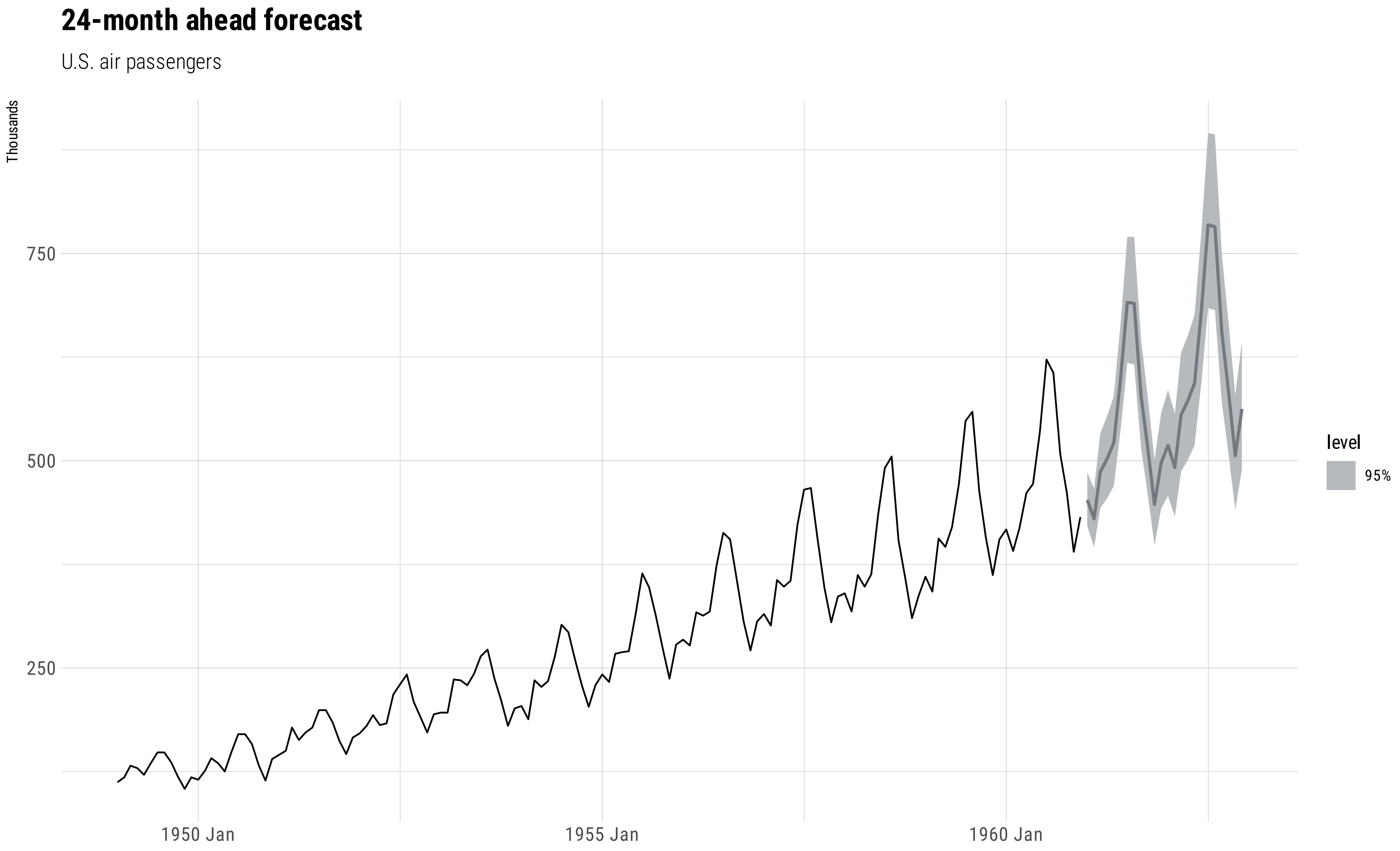
Seasonal ARIMA models

Are the residuals **white noise**?

```
air_log_fit |>
  augment() |>
  filter(.model == "arima_auto") |>
  features(.innov, ljung_box, lag = 2 * 12, dof = 3)
```

```
#> # A tibble: 1 × 3
#>   .model     lb_stat lb_pvalue
#>   <chr>        <dbl>     <dbl>
#> 1 arima_auto    29.0     0.113
```

Seasonal ARIMA models



Seasonal ARIMA models

```r
air_log_fit |>
  select(arima_auto) |>
  forecast(h = 24) |>
  head(5)
```

```
#> # A fable: 5 x 4 [1M]
#> # Key:     .model [1]
#>   .model         date        passengers .mean
#>   <chr>         <mth>            <dist> <dbl>
#> 1 arima_auto 1961 Jan t(N(6.1, 0.0013))  453.
#> 2 arima_auto 1961 Feb t(N(6.1, 0.0018))  430.
#> 3 arima_auto 1961 Mar t(N(6.2, 0.0022))  486.
#> 4 arima_auto 1961 Apr t(N(6.2, 0.0025))  502.
#> 5 arima_auto 1961 May t(N(6.3, 0.0028))  522.
```

Next time: Dynamic regression models