ADS 506 Final Team Project

Gagandeep Singh, Francisco Hernandez and Saloni Barhate

2025-12-03

```r
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)

# Core tidyverse tools used in the assignment
library(dplyr)        # %>%, count(), mutate(), slice_head(), slice_tail()
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)      # plotting for EDA

# Required for percent_format() in bivariate plots
library(scales)

# FPP3 Time Series Framework (tsibble, fable, feasts)
library(fpp3)
```

```
## Registered S3 method overwritten by 'tsibble':
##   method               from
##   as_tibble.grouped_df dplyr

## -- Attaching packages -------------------------------------------- fpp3 1.0.2 --

## v tibble      3.3.0     v tsibbledata 0.4.1
## v tidyr       1.3.1     v feasts      0.4.2
## v lubridate   1.9.4     v fable       0.4.1
## v tsibble     1.1.6

## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```r
# Neural Networks + NNETAR inside fable
library(fable.prophet)
```

```
## Loading required package: Rcpp
```

```r
library(tictoc)         # used for timing model fitting
```

```r
# Read the CSV file into an R data frame
births_df <- read.csv("daily-total-female-births-CA.csv") %>%
  mutate(
    date = as.Date(date, format = "%Y-%m-%d")  # parse the date string
  ) %>%
  rename(
    Date   = date,
    Births = births
  )

# Quick check
head(births_df)
```

```
##         Date Births
## 1 1959-01-01     35
## 2 1959-01-02     32
## 3 1959-01-03     30
## 4 1959-01-04     31
## 5 1959-01-05     44
## 6 1959-01-06     29
```

```r
str(births_df)
```

```
## 'data.frame':    365 obs. of  2 variables:
##  $ Date  : Date, format: "1959-01-01" "1959-01-02" ...
##  $ Births: int  35 32 30 31 44 29 45 43 38 27 ...
```

Initial Data Overview and Data Quality Check

```r
# Get the number of rows and columns
print("Dimensions of the data frame:")
```

```
## [1] "Dimensions of the data frame:"
```

```r
dim(births_df)
```

```
## [1] 365   2
```

```r
# List all column names
print("Column names:")
```

```
## [1] "Column names:"
```

```r
names(births_df)
```

```
## [1] "Date"   "Births"
```

```r
# Get a statistical summary of each column
print("Statistical summary of each column:")
```

```
## [1] "Statistical summary of each column:"
```

```r
summary(births_df)
```

```
##       Date                Births
##  Min.   :1959-01-01   Min.   :23.00
```

```
##  1st Qu.:1959-04-02   1st Qu.:37.00
##  Median :1959-07-02   Median :42.00
##  Mean   :1959-07-02   Mean   :41.98
##  3rd Qu.:1959-10-01   3rd Qu.:46.00
##  Max.   :1959-12-31   Max.   :73.00
```

```r
# Display the structure of the data frame (already done in previous cell, but re-confirming as per inst
print("Structure of the data frame:")
```

```
## [1] "Structure of the data frame:"
```

```r
print(str(births_df))
```

```
## 'data.frame':    365 obs. of  2 variables:
##  $ Date  : Date, format: "1959-01-01" "1959-01-02" ...
##  $ Births: int  35 32 30 31 44 29 45 43 38 27 ...
## NULL
```

Handle Missing Values

```r
# 1. Calculate total number of missing values (NA) for each column
missing_values_count <- colSums(is.na(births_df))
missing_values_count
```

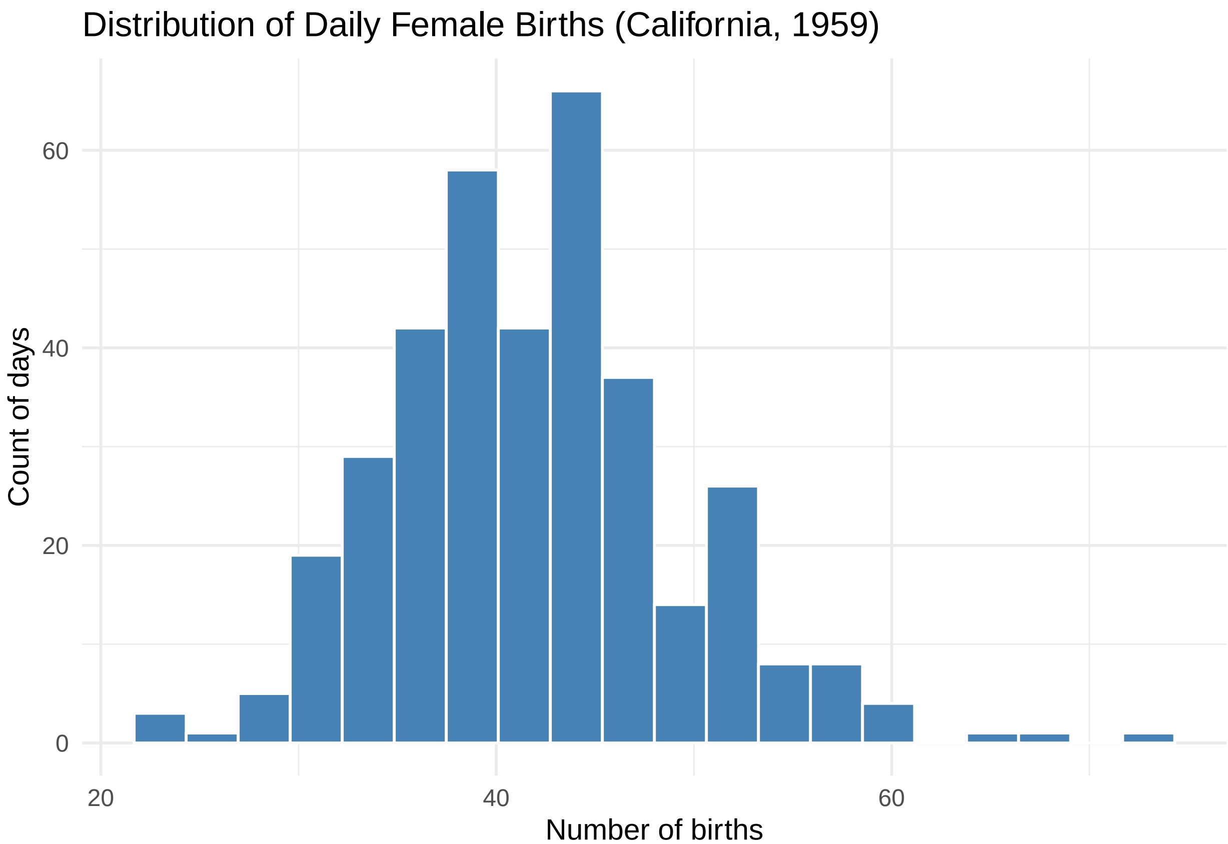
```
##   Date Births
##      0      0
```

```r
# Percentage of missing values

missing_values_percentage <- (missing_values_count / nrow(births_df)) * 100
data.frame(
Column = names(missing_values_count),
NACount = missing_values_count,
NAPercentage = missing_values_percentage
)
```

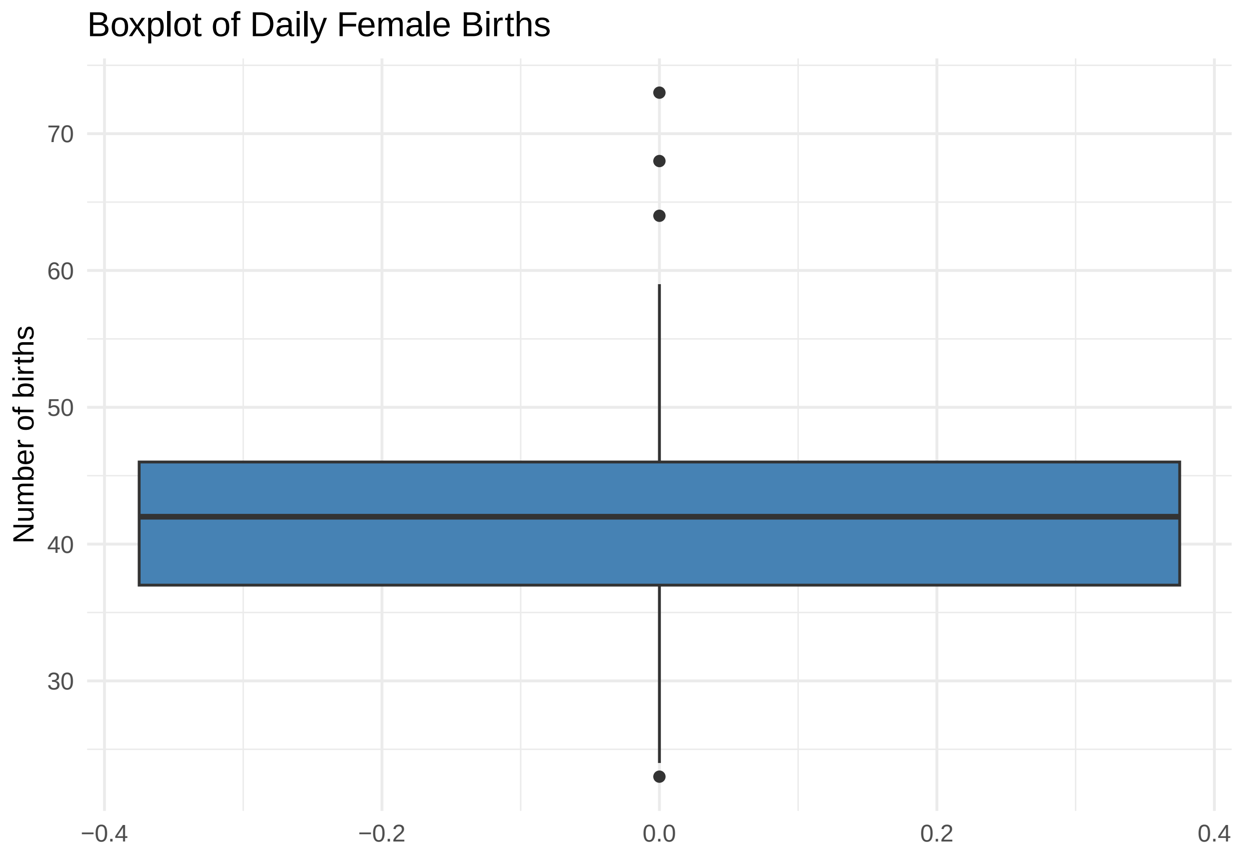
```
##        Column NACount NAPercentage
## Date     Date       0            0
## Births Births       0            0
```

Identify and Address Outliers

```r
# Histogram of daily births

ggplot(births_df, aes(x = Births)) +
geom_histogram(bins = 20, fill = "steelblue", color = "white") +
labs(
title = "Distribution of Daily Female Births (California, 1959)",
x = "Number of births",
y = "Count of days"
) +
theme_minimal()
```




```r
# Boxplot to check for outliers

ggplot(births_df, aes(y = Births)) +
geom_boxplot(fill = "steelblue") +
labs(
title = "Boxplot of Daily Female Births",
y = "Number of births"
) +
theme_minimal()
```



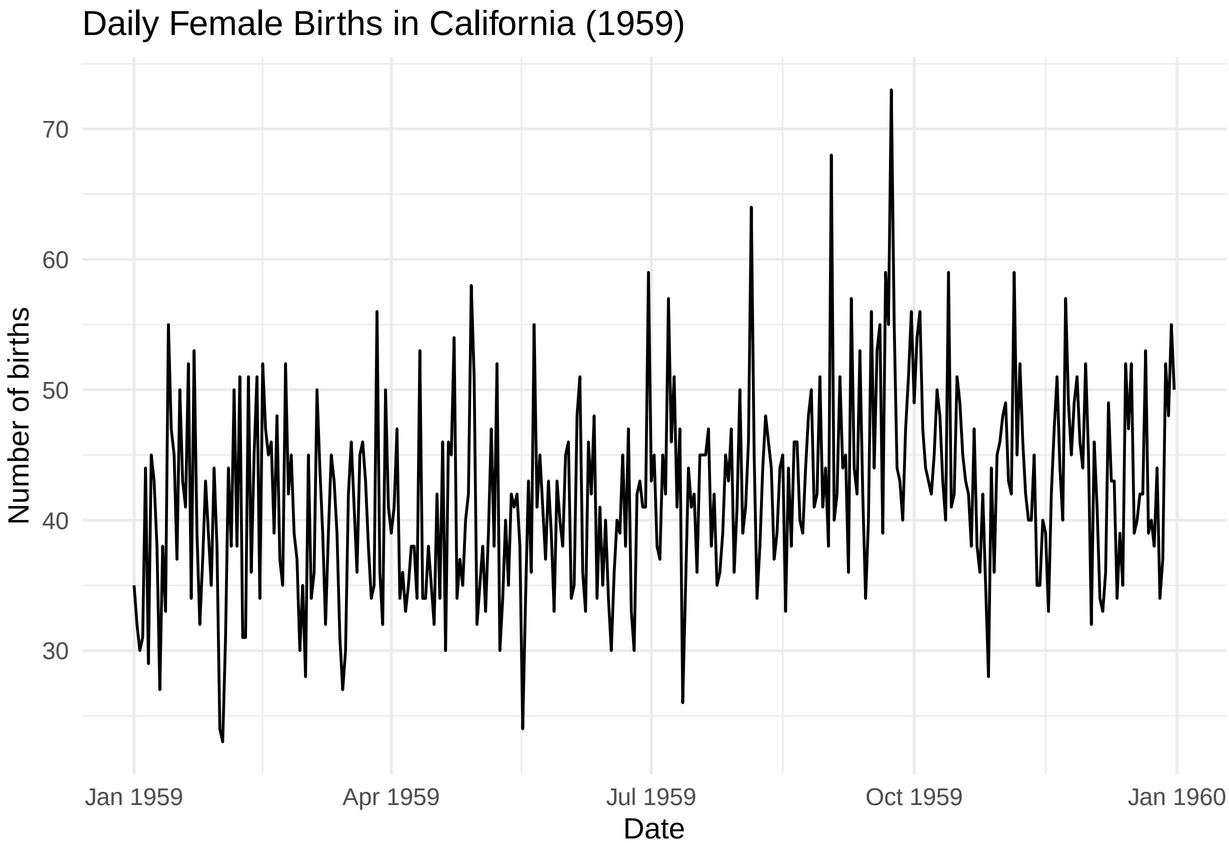
Data Type Conversion and Transformation

```
# Ensure Date is a proper Date class

births_df$Date <- as.Date(births_df$Date)

# Convert to tsibble

births_ts <- births_df %>%
as_tsibble(index = Date)

births_ts %>% head()
```

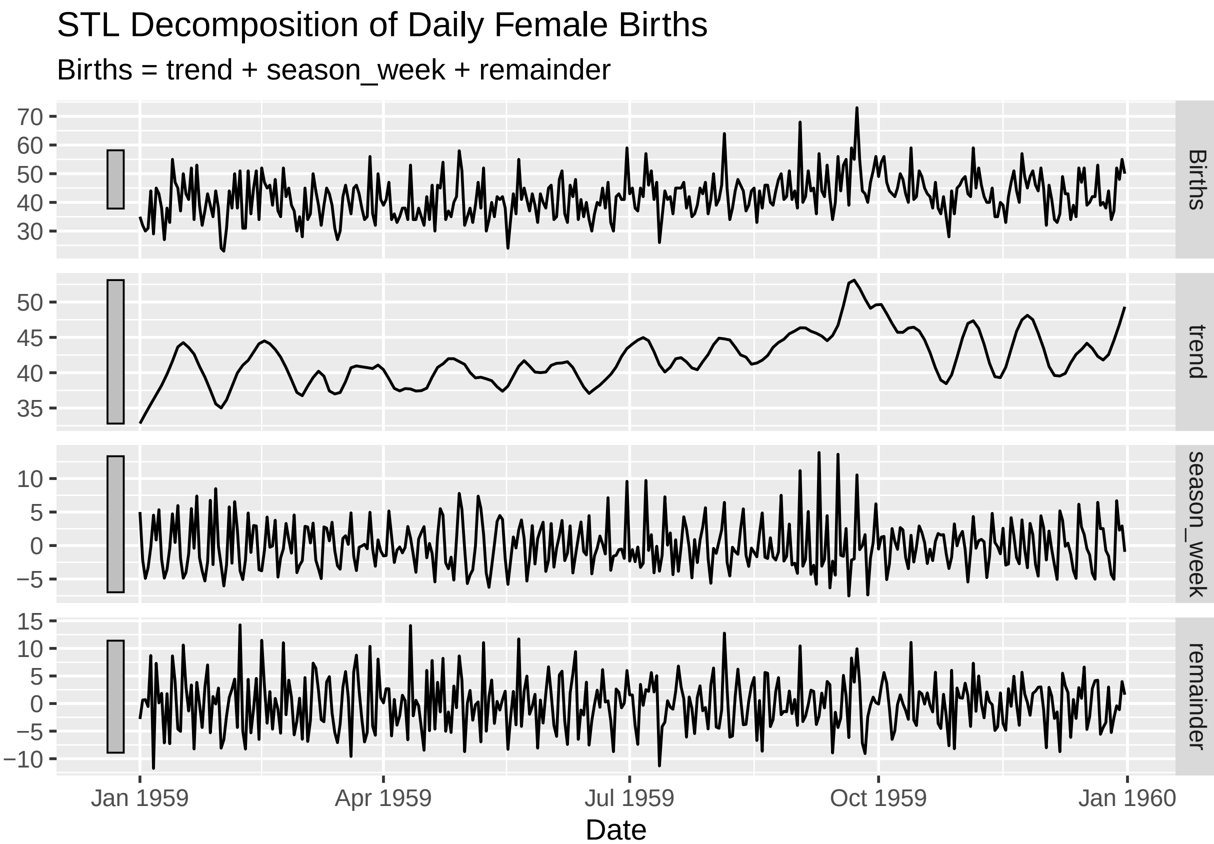
```
## # A tsibble: 6 x 2 [1D]
##   Date       Births
##   <date>      <int>
## 1 1959-01-01     35
## 2 1959-01-02     32
## 3 1959-01-03     30
## 4 1959-01-04     31
## 5 1959-01-05     44
## 6 1959-01-06     29
```

```
# Plot the time series

autoplot(births_ts,  Births) +
labs(
```

```r
title = "Daily Female Births in California (1959)",
x = "Date",
y = "Number of births"
) +
theme_minimal()
```

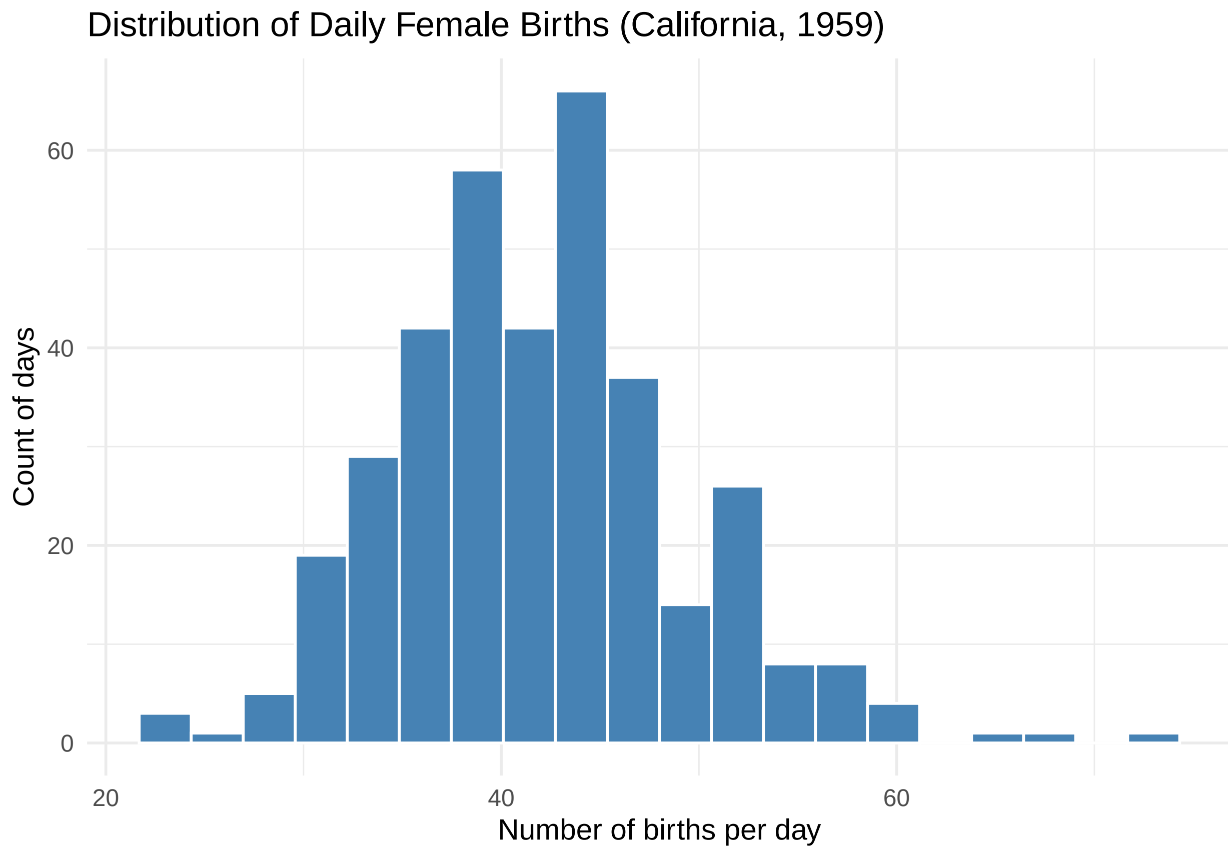



```r
births_ts %>%
model(STL(Births ~ season(window = 7))) %>%
components() %>%
autoplot() +
labs(
title = "STL Decomposition of Daily Female Births",
x = "Date"
)
```

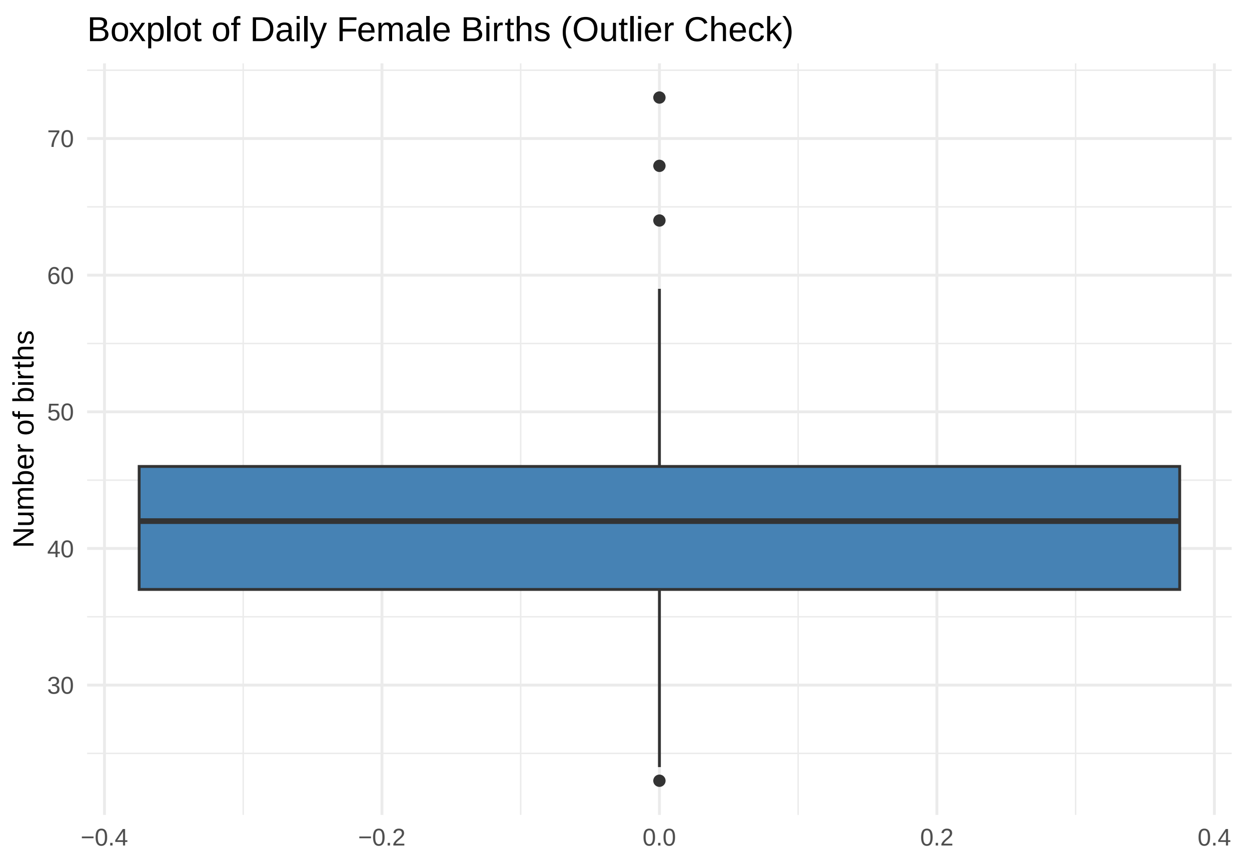


Univariate Exploration

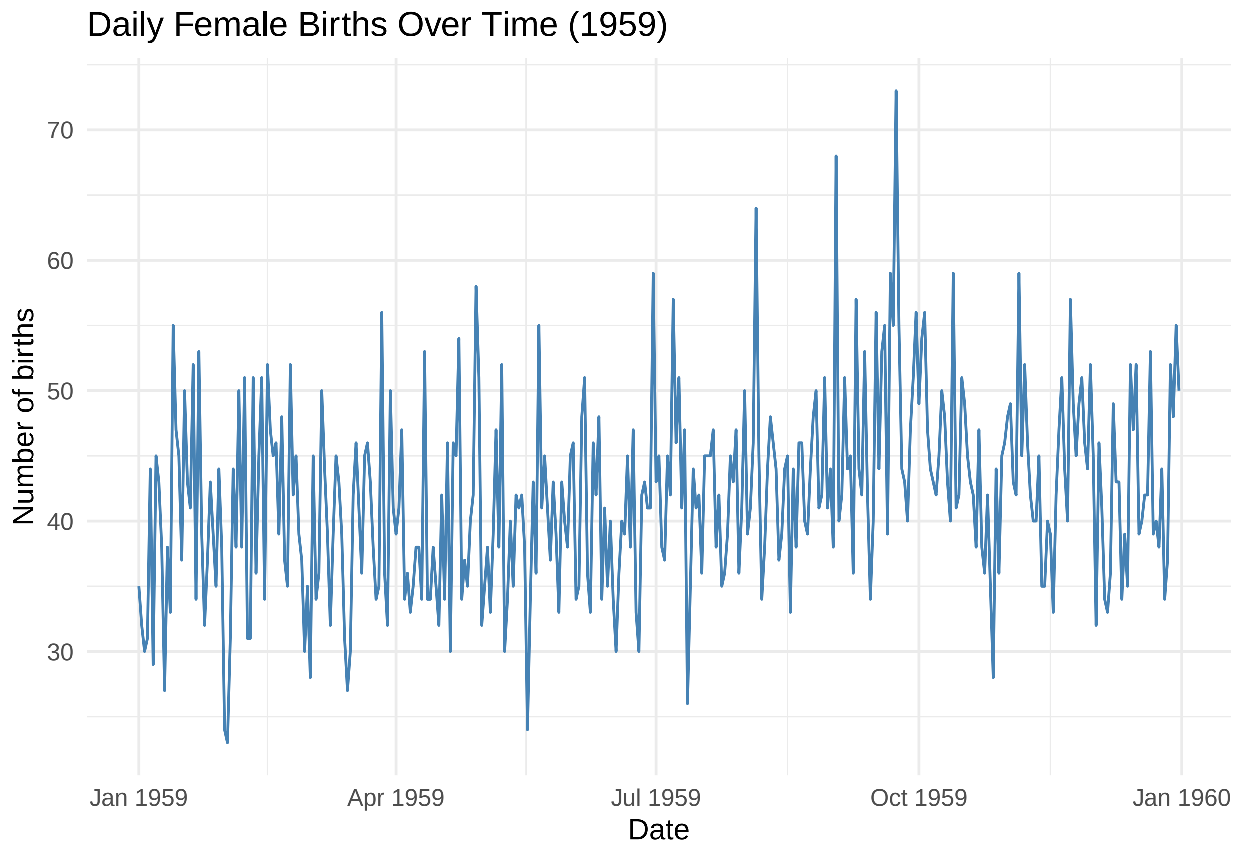
```
## Univariate Exploration

#-----------------------------------------------------------
# Distribution of daily births
#-----------------------------------------------------------
births_df %>%
  ggplot(aes(x = Births)) +
  geom_histogram(bins = 20, fill = "steelblue", color = "white") +
  labs(
    title = "Distribution of Daily Female Births (California, 1959)",
    x = "Number of births per day",
    y = "Count of days"
  ) +
  theme_minimal()
```




```r
#-----------------------------------------------------------
# Boxplot for detecting potential outliers
#-----------------------------------------------------------
births_df %>%
  ggplot(aes(y = Births)) +
  geom_boxplot(fill = "steelblue") +
  labs(
    title = "Boxplot of Daily Female Births (Outlier Check)",
    y = "Number of births"
  ) +
  theme_minimal()
```

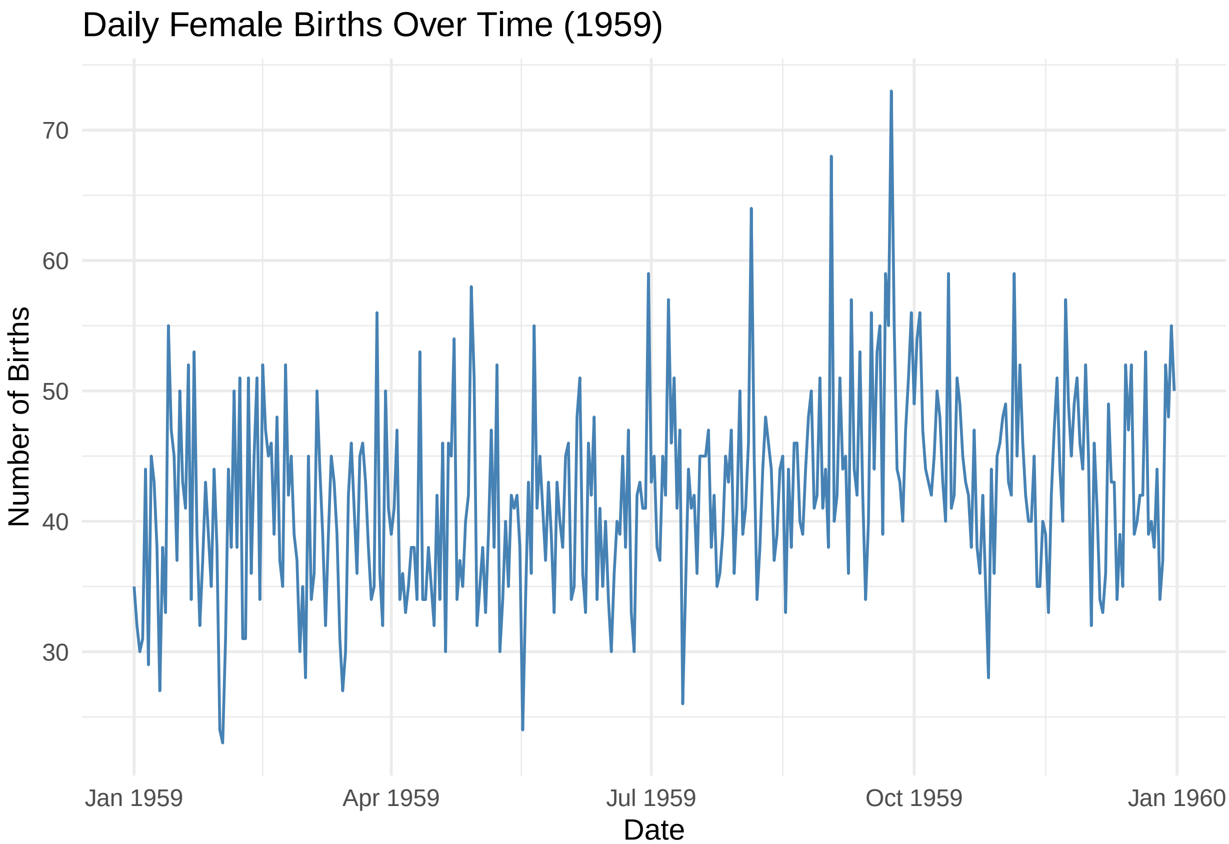



```r
#-----------------------------------------------------------
# Time plot before creating tsibble (raw form)
#-----------------------------------------------------------
ggplot(births_df, aes(x = Date, y = Births)) +
  geom_line(color = "steelblue") +
  labs(
    title = "Daily Female Births Over Time (1959)",
    x = "Date",
    y = "Number of births"
  ) +
  theme_minimal()
```

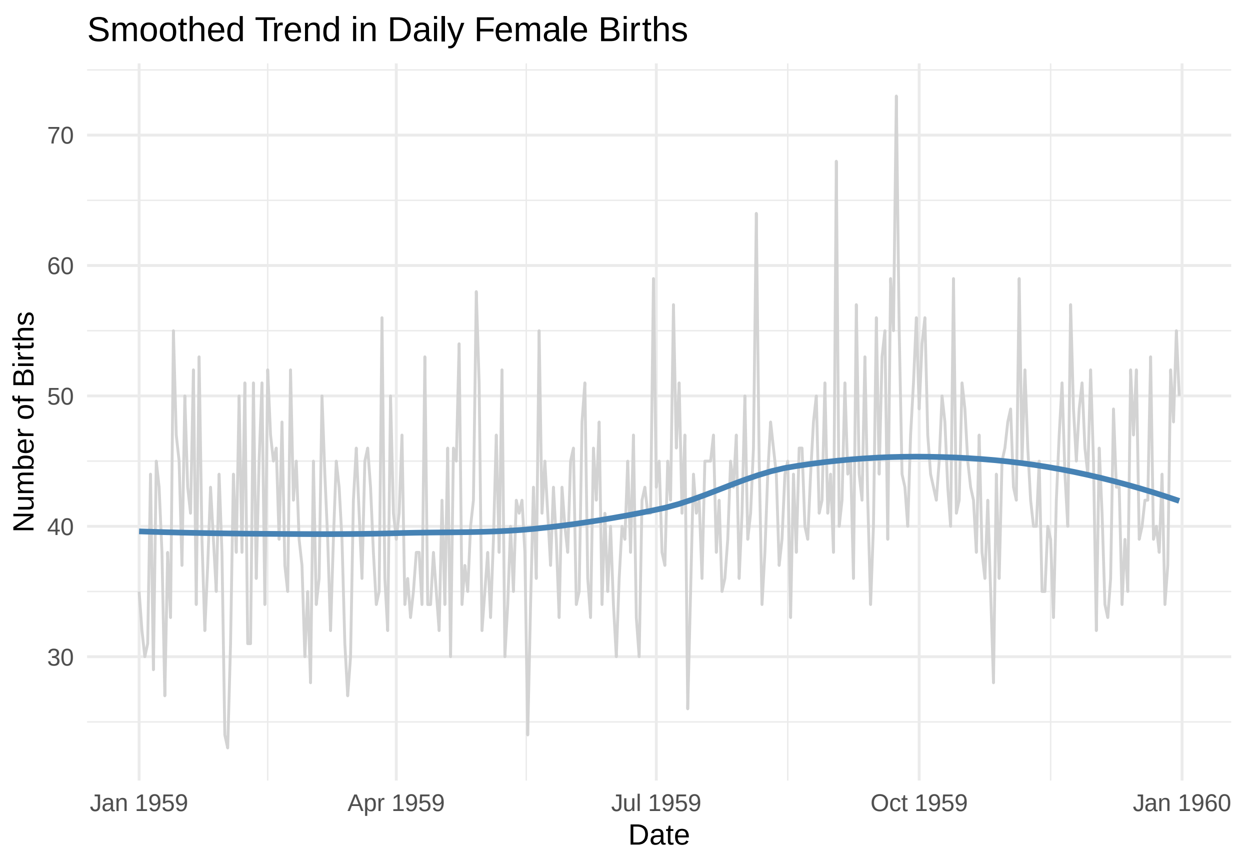


Bivariate Exploration

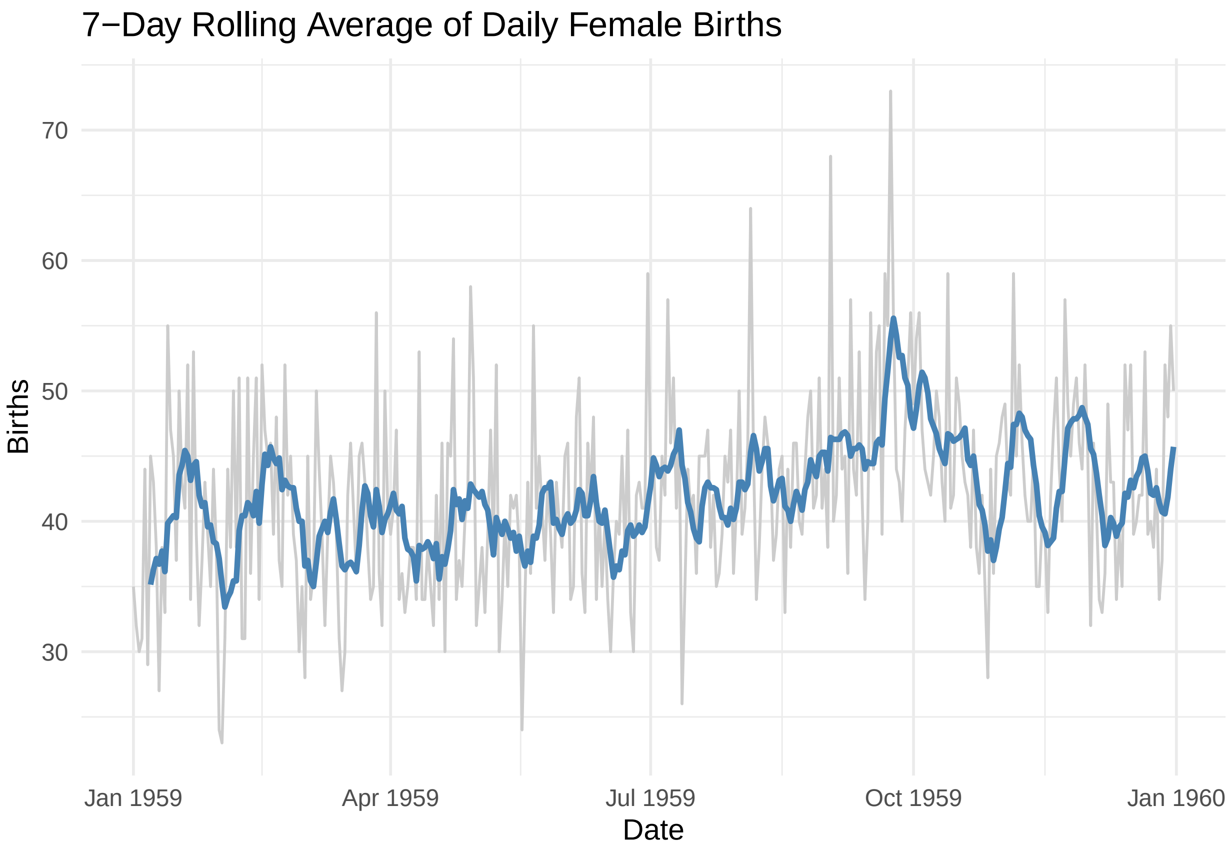
```
## Bivariate Exploration

#----------------------------------------------------------
# Births over time (line plot)
#----------------------------------------------------------
ggplot(births_df, aes(x = Date, y = Births)) +
  geom_line(color = "steelblue") +
  labs(
    title = "Daily Female Births Over Time (1959)",
    x = "Date",
    y = "Number of Births"
  ) +
  theme_minimal()
```



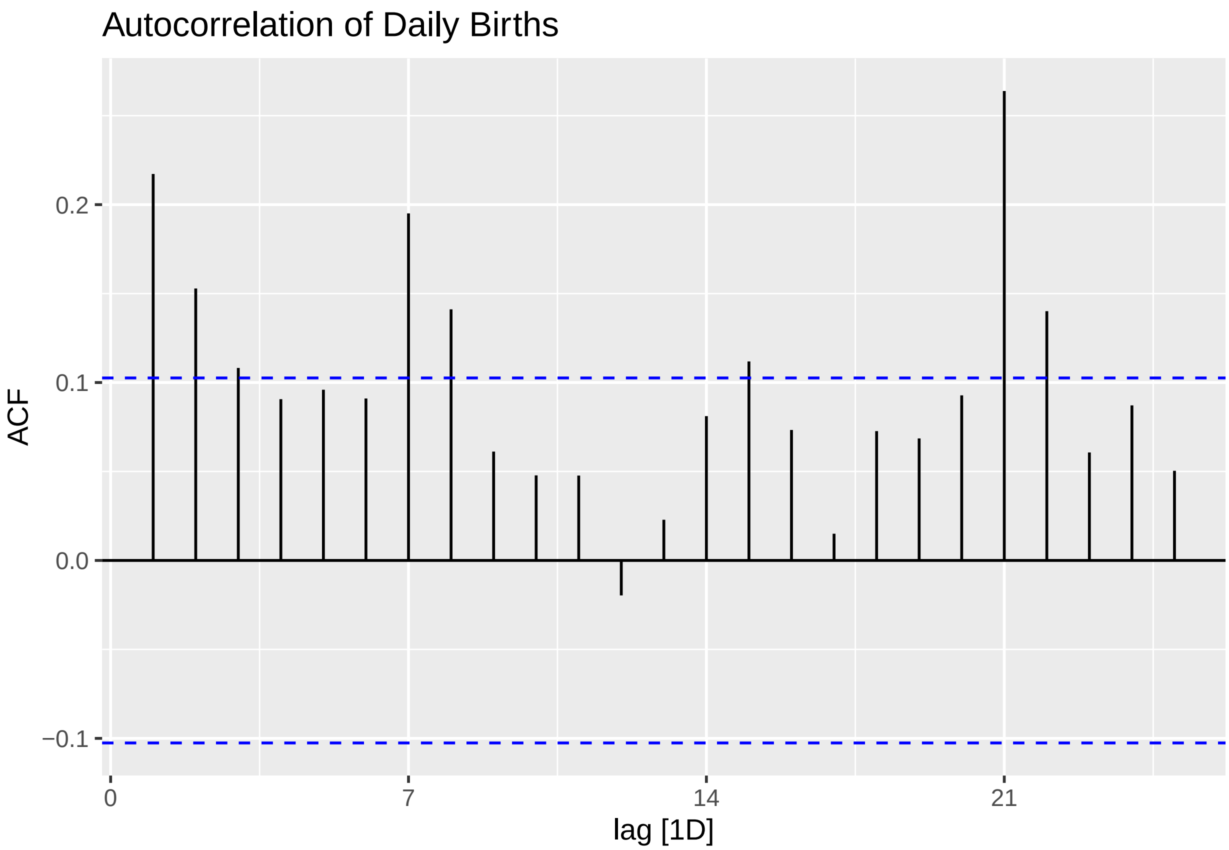

```r
#-----------------------------------------------------------
# Births with smoothing line (reveals underlying pattern)
#-----------------------------------------------------------
ggplot(births_df, aes(x = Date, y = Births)) +
  geom_line(color = "lightgray") +
  geom_smooth(method = "loess", se = FALSE, color = "steelblue") +
  labs(
    title = "Smoothed Trend in Daily Female Births",
    x = "Date",
    y = "Number of Births"
  ) +
  theme_minimal()
```



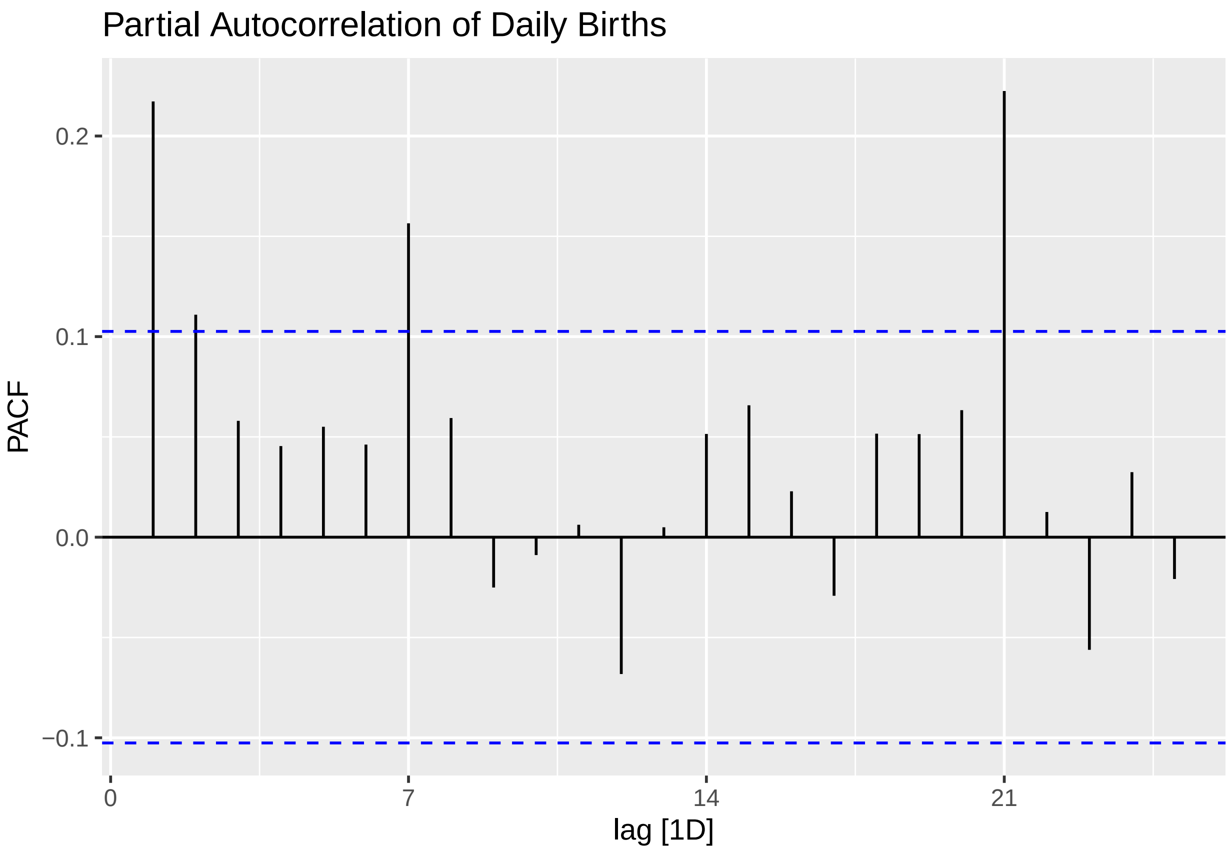

```r
#------------------------------------------------------------
# 7-day rolling mean (common in time series EDA)
#------------------------------------------------------------
births_df %>%
  mutate(Rolling7 = slider::slide_dbl(Births, mean, .before = 6, .complete = TRUE)) %>%
  ggplot(aes(x = Date)) +
  geom_line(aes(y = Births), color = "gray80") +
  geom_line(aes(y = Rolling7), color = "steelblue", size = 1) +
  labs(
    title = "7-Day Rolling Average of Daily Female Births",
    x = "Date",
    y = "Births"
  ) +
  theme_minimal()
```




```r
#-----------------------------------------------------------
# Autocorrelation and partial autocorrelation (time-based relationship)
#-----------------------------------------------------------
births_ts %>%
  ACF(Births) %>%
  autoplot() +
  labs(title = "Autocorrelation of Daily Births", y = "ACF")
```




```
births_ts %>%
  PACF(Births) %>%
  autoplot() +
  labs(title = "Partial Autocorrelation of Daily Births", y = "PACF")
```



Outlier and Data Quality Checks

```r
summary(births_df)
```

```
##       Date                Births
##  Min.   :1959-01-01   Min.   :23.00
##  1st Qu.:1959-04-02   1st Qu.:37.00
##  Median :1959-07-02   Median :42.00
##  Mean   :1959-07-02   Mean   :41.98
##  3rd Qu.:1959-10-01   3rd Qu.:46.00
##  Max.   :1959-12-31   Max.   :73.00
```

```r
# Check for remaining NA values
colSums(is.na(births_df))
```

```
##   Date Births
##      0      0
```

```r
#Check duplicates
sum(duplicated(births_df))
```

```
## [1] 0
```

Train / validation split

```r
train_end <- as.Date("1959-10-31")

births_train <- births_ts %>%
```

```r
filter(Date <= train_end)

births_valid <- births_ts %>%
filter(Date > train_end)

cat("Number of observations TOTAL:      ", nrow(births_ts), "\n")
```

```
## Number of observations TOTAL:       365
```

```r
cat("Number of observations in TRAIN:   ", nrow(births_train), "\n")
```

```
## Number of observations in TRAIN:    304
```

```r
cat("Number of observations in VALIDATION:", nrow(births_valid), "\n")
```

```
## Number of observations in VALIDATION: 61
```

Fit ARIMA, ETS, and NNETAR to TreatmentRate

```r
#=========================================================

# Fit ARIMA, ETS, NNETAR on Births (training data)

#=========================================================

tic("Fit ARIMA + ETS + NNETAR (Births)")

births_models <- births_train %>%
model(
ARIMA  = ARIMA(Births),
ETS    = ETS(Births),
NNETAR = NNETAR(Births, repeats = 20)
)

toc()
```

```
## Fit ARIMA + ETS + NNETAR (Births): 2.927 sec elapsed
```

```r
births_models
```

```
## # A mable: 1 x 3
##                       ARIMA          ETS             NNETAR
##                     <model>      <model>            <model>
## 1 <ARIMA(0,1,1)(2,0,0)[7]> <ETS(A,N,A)> <NNAR(16,1,8)[7]>
```

```r
births_models %>% select(ARIMA)  %>% report()
```

```
## Series: Births
## Model: ARIMA(0,1,1)(2,0,0)[7]
##
## Coefficients:
##           ma1    sar1     sar2
##       -0.9491  0.1290  -0.0286
## s.e.   0.0269  0.0602   0.0595
##
## sigma^2 estimated as 50.99:  log likelihood=-1025.22
## AIC=2058.45   AICc=2058.58   BIC=2073.3
```

```r
births_models %>% select(ETS)    %>% report()
```

```
## Series: Births
## Model: ETS(A,N,A)
##   Smoothing parameters:
##     alpha = 0.06260596
##     gamma = 0.0001263315
##
##   Initial states:
##      l[0]     s[0]    s[-1]     s[-2]     s[-3]      s[-4]     s[-5]     s[-6]
##  38.33528 2.179863 2.422009 -2.040249 -3.141018 -0.3772689 0.2529517 0.7037109
##
##   sigma^2:  49.0394
##
##      AIC     AICc      BIC
## 2932.198 2932.949 2969.368
```

```r
births_models %>% select(NNETAR) %>% report()
```

```
## Series: Births
## Model: NNAR(16,1,8)[7]
##
## Average of 20 networks, each of which is
## a 16-8-1 network with 145 weights
## options were - linear output units
##
## sigma^2 estimated as 2.616
```

```r
# Forecast horizon = number of validation observations

h_valid <- nrow(births_valid)

# Generate forecasts from all three models

fc_valid <- births_models %>%
forecast(h = h_valid)

# Compute accuracy on held-out validation data

births_acc <- fc_valid %>%
accuracy(births_valid) %>%
select(.model, RMSE, MAE, MAPE) %>%
arrange(RMSE)

births_acc
```

```
## # A tibble: 3 x 4
##   .model  RMSE   MAE  MAPE
##   <chr>  <dbl> <dbl> <dbl>
## 1 ARIMA   6.48  5.35  12.6
## 2 ETS     6.67  5.39  12.5
## 3 NNETAR  7.75  6.29  13.7
```

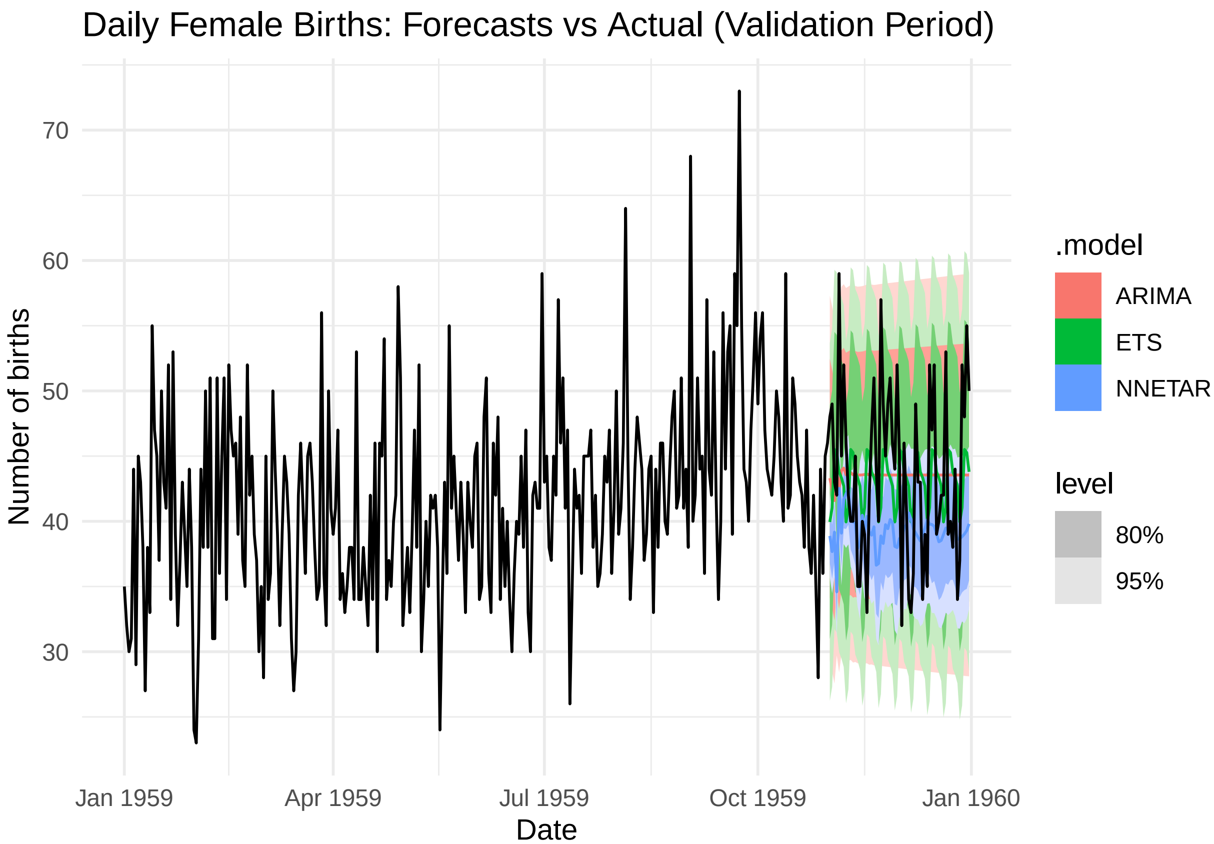
```r
knitr::kable(
births_acc,
digits = 4,
```

```
caption = "Validation Accuracy for Daily Female Births (ARIMA, ETS, NNETAR)"
)
```

Table 1: Validation Accuracy for Daily Female Births (ARIMA, ETS, NNETAR)

.model	RMSE	MAE	MAPE
ARIMA	6.4810	5.3522	12.5721
ETS	6.6653	5.3903	12.4964
NNETAR	7.7512	6.2944	13.6642

```
fc_valid %>%
mutate(
Model = case_when(
.model == "ARIMA"   ~ "ARIMA(Births)",
.model == "ETS"     ~ "ETS(Births)",
.model == "NNETAR"  ~ "NNETAR(Births)"
)
) %>%
autoplot(data = births_ts) +
labs(
title = "Daily Female Births: Forecasts vs Actual (Validation Period)",
x = "Date",
y = "Number of births"
) +
theme_minimal()
```


Daily Female Births: Forecasts vs Actual (Validation Period)
Number of births
70
60
50
40
30
Jan 1959
Apr 1959
Jul 1959
Oct 1959
Jan 1960
Date
.model
ARIMA
ETS
NNETAR
level
80%
95%