Week 9: In Class

Mitchell Levy

Data Visualization

10/25/2024

Load tidyverse.

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to b
```

```
library(lubridate)
library(dplyr)
```

Look at data.

```
read_csv("weatherAUS.csv") -> weather
```

```
Rows: 145460 Columns: 23
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr   (6): Location, WindGustDir, WindDir9am, WindDir3pm, RainToday, RainTom...
dbl  (16): MinTemp, MaxTemp, Rainfall, Evaporation, Sunshine, WindGustSpeed,...
date  (1): Date

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
weather
```

Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine
<date>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2008-12-01	Albury	13.4	22.9	0.6	NA	NA
2008-12-02	Albury	7.4	25.1	0.0	NA	NA
2008-12-03	Albury	12.9	25.7	0.0	NA	NA
2008-12-04	Albury	9.2	28.0	0.0	NA	NA
2008-12-05	Albury	17.5	32.3	1.0	NA	NA
2008-12-06	Albury	14.6	29.7	0.2	NA	NA
2008-12-07	Albury	14.3	25.0	0.0	NA	NA
2008-12-08	Albury	7.7	26.7	0.0	NA	NA
2008-12-09	Albury	9.7	31.9	0.0	NA	NA
2008-12-10	Albury	13.1	30.1	1.4	NA	NA

1-10 of 10,000 rows | 1-7 of 23 columns   Previous 1 2 3 4 5 6 ... 1000 Next

1. For my location I am choosing “WaggaWagga” because the name is hilarious. For my year I am choosing 2011, because I like the number 11.

Let’s filter the data so that we only get WaggaWagga and 2011.

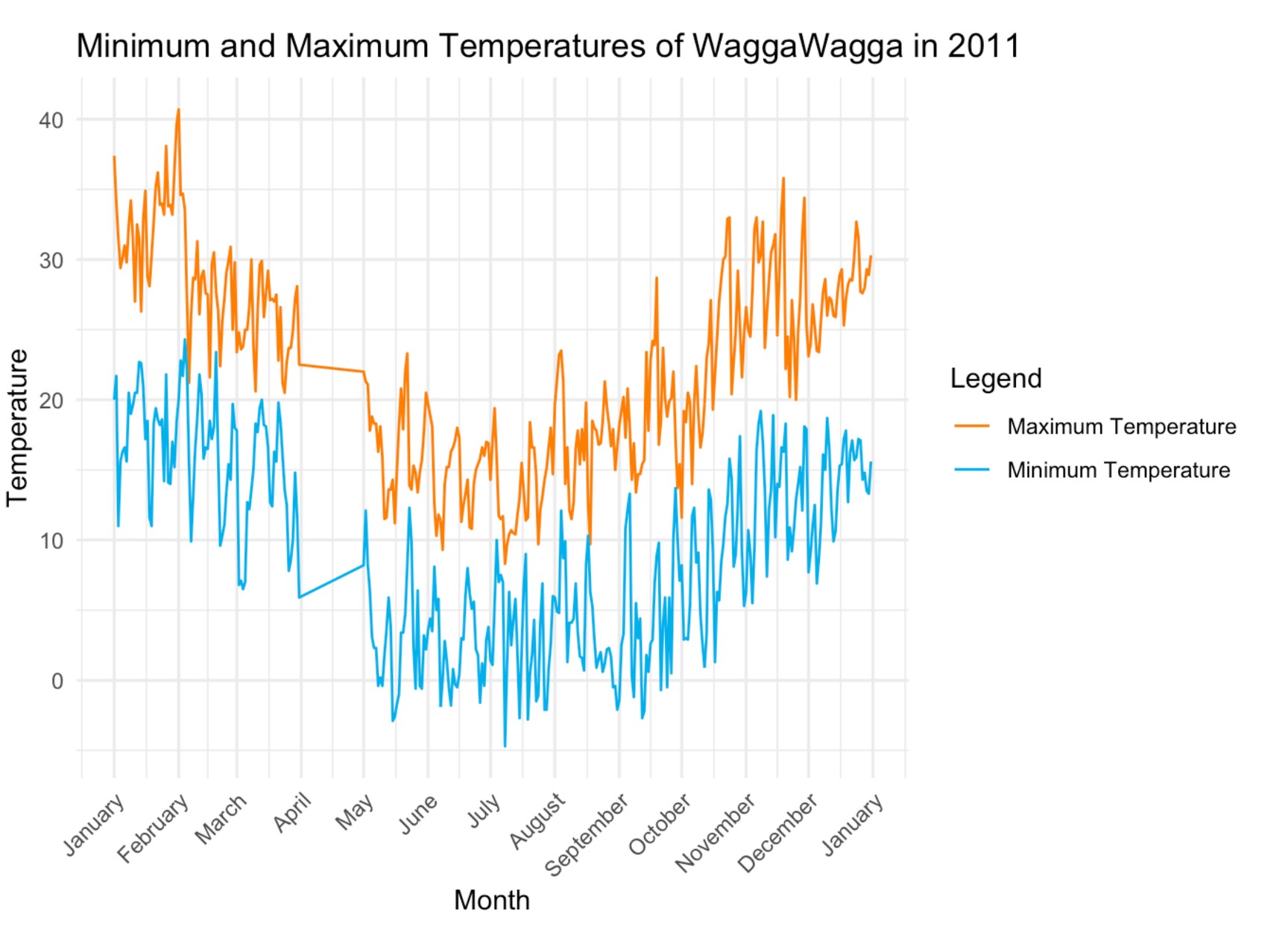
```
weather |>
  filter(Location == "WaggaWagga") |>
  filter(Date >= ymd(20110101) & Date < ymd(20120101)) -> weather.sub # Only want 2011
weather.sub
```

Date	Location	MinTemp	MaxTe...	Rainfall	Evaporation	Sunshine
<date>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2011-01-01	WaggaWagga	20.0	37.4	0.0	8.6	9.0
2011-01-02	WaggaWagga	21.7	34.1	0.0	7.4	10.8
2011-01-03	WaggaWagga	11.0	31.5	0.0	9.8	13.5
2011-01-04	WaggaWagga	15.6	29.4	0.0	9.8	12.0
2011-01-05	WaggaWagga	16.3	30.1	0.0	8.0	9.2
2011-01-06	WaggaWagga	16.6	31.0	0.0	6.8	13.5
2011-01-07	WaggaWagga	15.6	29.8	0.0	9.8	7.3
2011-01-08	WaggaWagga	20.5	32.4	0.0	8.0	12.6
2011-01-09	WaggaWagga	19.0	34.2	0.0	8.6	10.3
2011-01-10	WaggaWagga	19.6	31.5	2.4	8.6	6.8

1-10 of 335 rows | 1-7 of 23 columns   Previous 1 2 3 4 5 6 ... 34 Next

2. Construct a time-series visualization for your location and year that shows the minimum and maximum temperatures for each day.

```
weather.sub |>
  ggplot() +
  geom_line(aes(x = Date, y = MaxTemp, color = "Maximum Temperature")) +
  geom_line(aes(Date, MinTemp, color = "Minimum Temperature")) +
  ggtitle("Minimum and Maximum Temperatures of WaggaWagga in 2011") +
  labs(y = "Temperature", x = "Month", color = "Legend") +
  scale_color_manual(values = c("Maximum Temperature" = "darkorange1", "Minimum Temperat
  scale_x_date(date_labels = "%B", date_breaks = "1 month") +  # Show only months
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Tilt month names by 45 deg
```



3. Facet this visualization for multiple years (Maybe 4 in a row)

We need to make a new data frame so that we have all of the years have their own column.

```
weather |>
  drop_na(MinTemp) |> # get rid of nas
  drop_na(MaxTemp) |>
  mutate(Year = year(Date)) |> # make year col
  mutate(Month = month(Date)) |> # make month col
  group_by(Month, Year) |>
  summarise(MaxTempAvg = mean(MaxTemp), MinTempAvg = mean(MinTemp)) -> weather.date # ge
```

```
`summarise()` has grouped output by 'Month'. You can override using the
`.groups` argument.
```

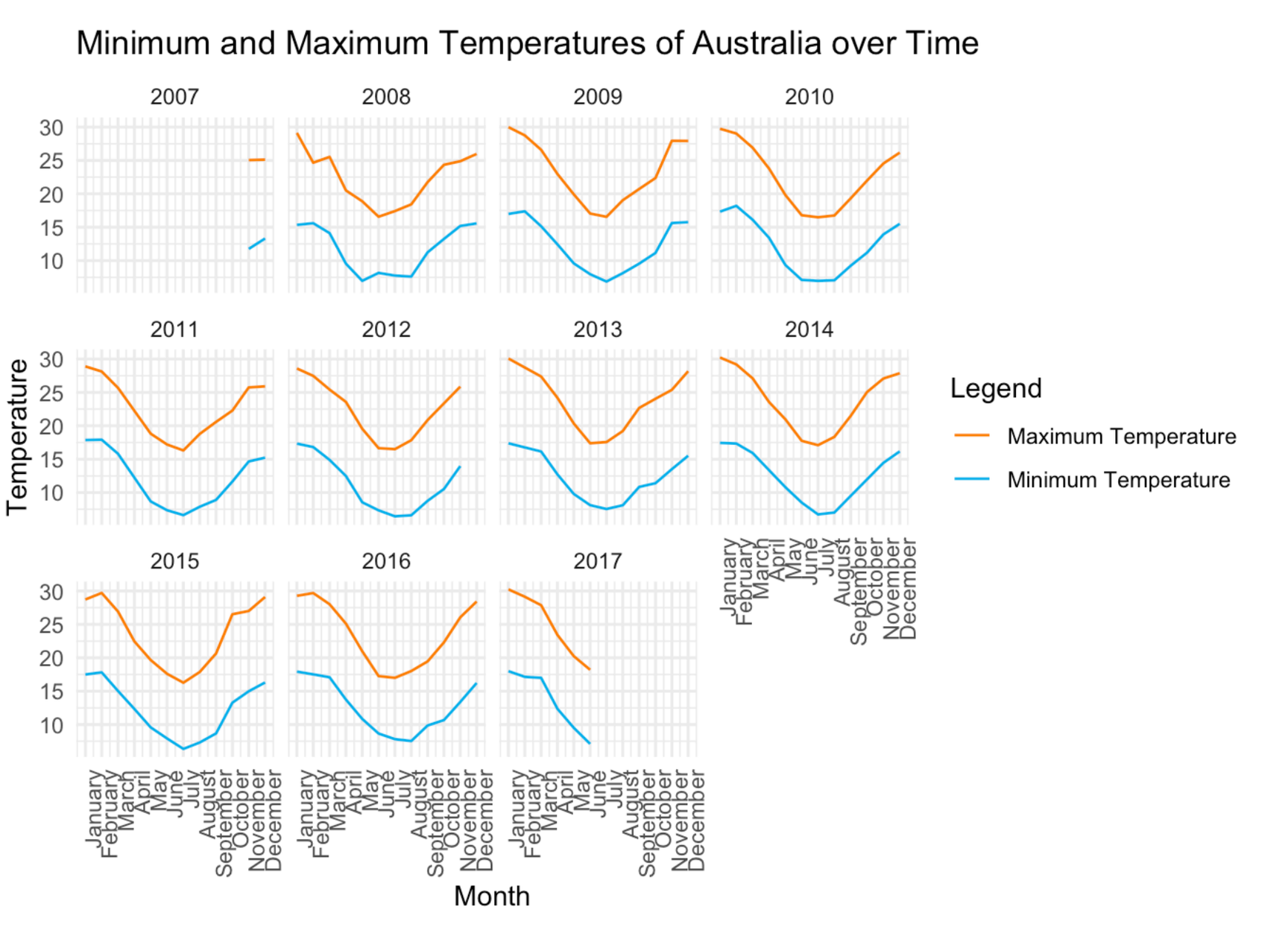
```
weather.date
```

Month	Year	MaxTempAvg	MinTempAvg
<dbl>	<dbl>	<dbl>	<dbl>
1	2008	29.12581	15.348387
1	2009	29.98256	16.992764
1	2010	29.74873	17.333499
1	2011	28.89384	17.868365
1	2012	28.58594	17.337385
1	2013	30.07516	17.385674
1	2014	30.23016	17.444674
1	2015	28.73188	17.506376
1	2016	29.27720	17.931741
1	2017	30.24423	18.001134

1-10 of 113 rows   Previous 1 2 3 4 5 6 ... 12 Next

Now we can make graphs with facet wrap.

```
weather.date |>
  ggplot() +
  geom_line(aes(x = Month, y = MaxTempAvg, color = "Maximum Temperature")) +
  geom_line(aes(x = Month, y = MinTempAvg, color = "Minimum Temperature")) +
  facet_wrap(~Year, ncol = 4) +
  ggtitle("Minimum and Maximum Temperatures of Australia over Time") +
  labs(y = "Temperature", color = "Legend") +
  scale_color_manual(values = c("Maximum Temperature" = "darkorange1", "Minimum Temperat
  scale_x_continuous(breaks = 1:12, labels = month.name) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Tilt month names by 45 deg
```

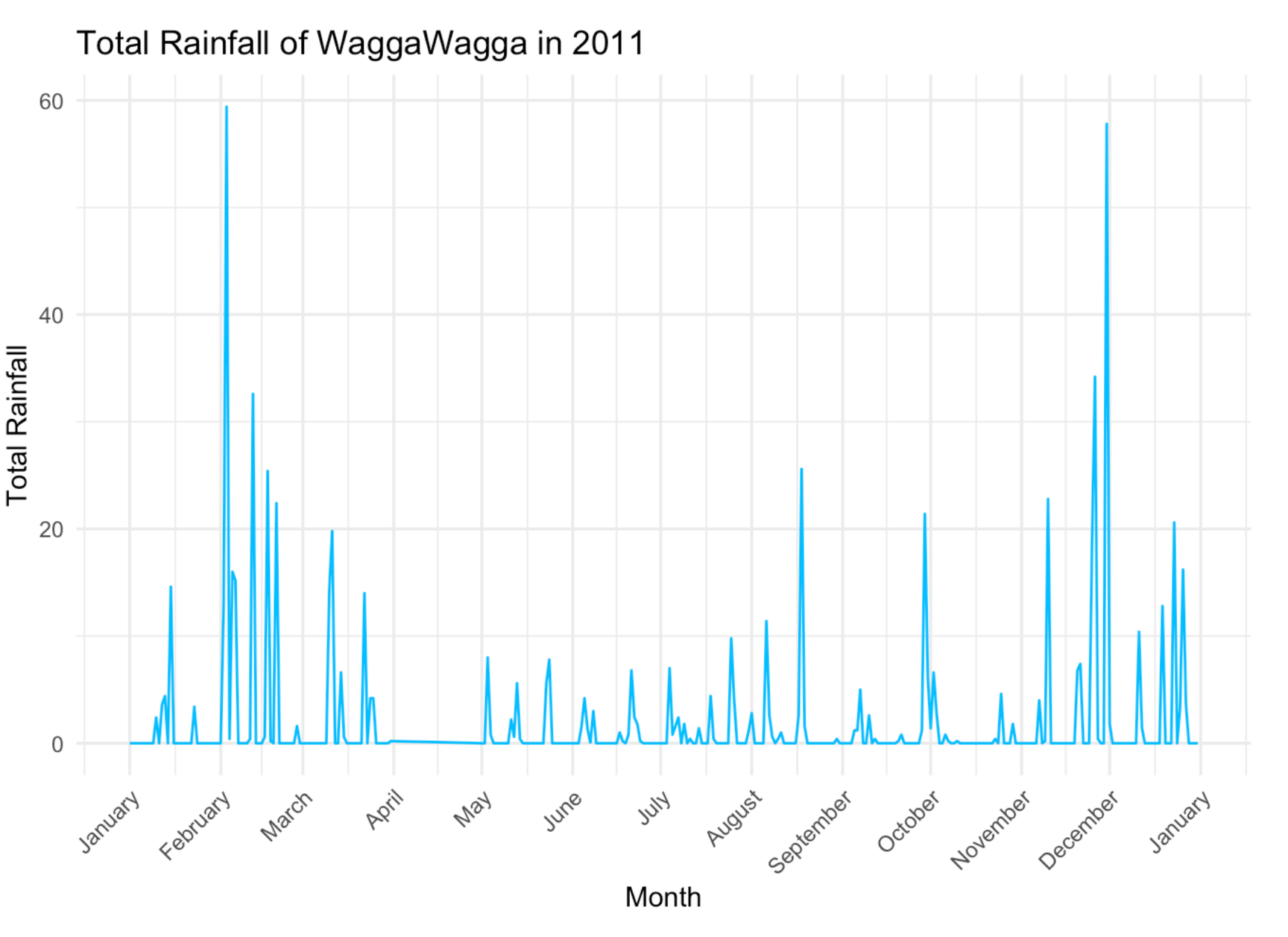


4. Comment on your faceted visualization.

This graph shows the minimum and maximum average temperatures per month in Australia over time. The orange line is max temp, and the blue line is min temp/ The axis show the months. What is good about this graph is that it is easy to read and see the trends of the graph. For example, June and July always have the lowest temps every year. I wish I could figure out how to make this graph bigger so that the month names could be easier to read, but other than that, this graph is simple and effective.

5. Construct a time-series visualization for your Location and year that shows total rainfall in each month.

```
weather.sub |>
  ggplot() +
  geom_line(aes(x = Date, y = Rainfall, color = "rainfall")) +
  ggtitle("Total Rainfall of WaggaWagga in 2011") +
  labs(y = "Total Rainfall", x = "Month", color = "Legend") +
  scale_color_manual(values = c("rainfall" = "deepskyblue1")) +
  scale_x_date(date_labels = "%B", date_breaks = "1 month") +  # Show only months
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) + # Tilt month names by 45 de
  theme(legend.position = "none")
```



6. Facet this visualization for multiple years. (Maybe 4 in a row)

```
weather |>
  drop_na(Rainfall) |> # get rid of nas
  mutate(Year = year(Date)) |> # make year col
  mutate(Month = month(Date)) |> # make month col
  group_by(Month, Year) |>
  summarise(RainAvg = mean(Rainfall)) -> weather.rain # get averages of each month
```

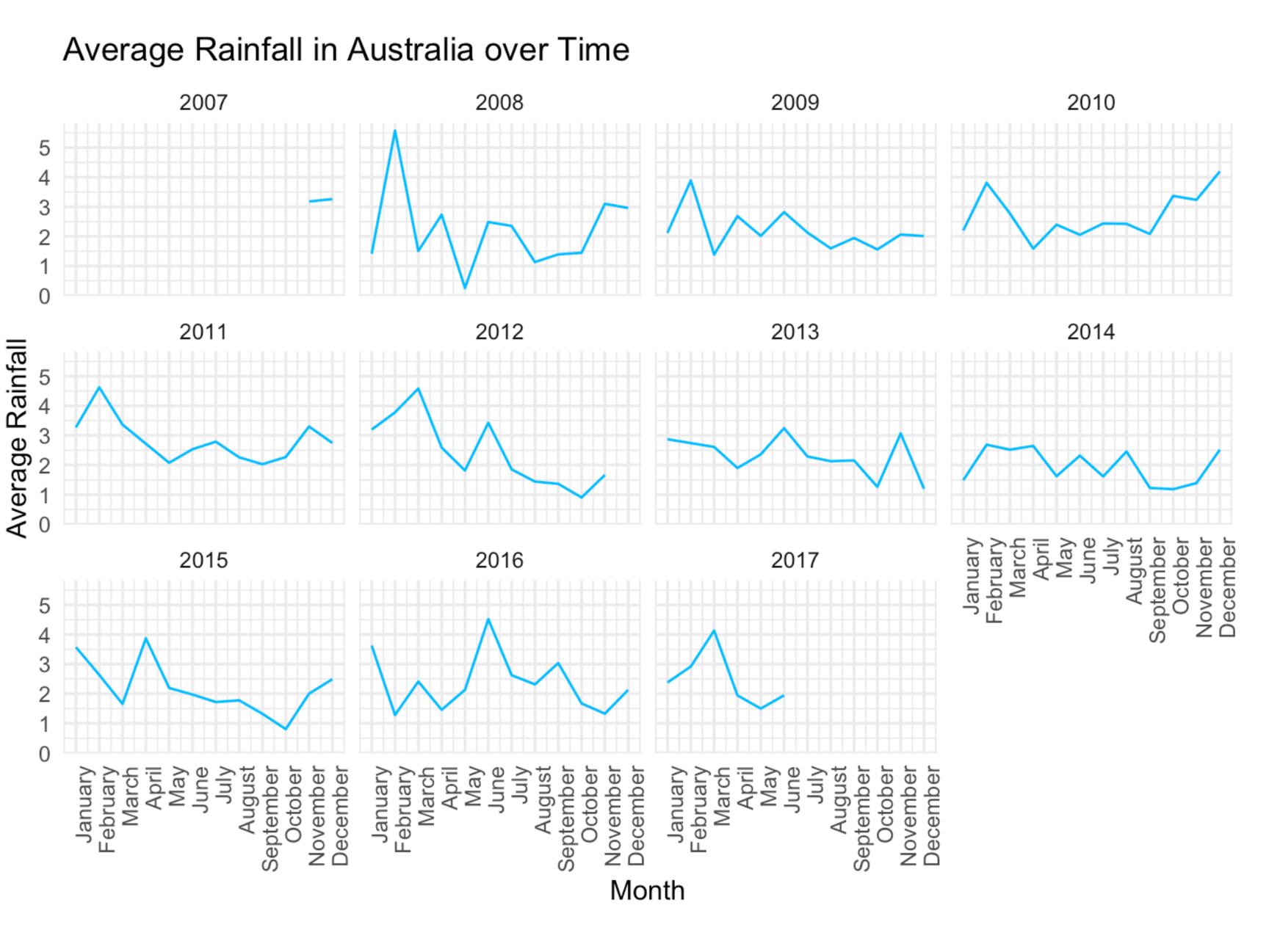
```
`summarise()` has grouped output by 'Month'. You can override using the
`.groups` argument.
```

```
weather.rain
```

Month	Year	RainAvg
<dbl>	<dbl>	<dbl>
1	2008	1.4129032
1	2009	2.1185212
1	2010	2.1982918
1	2011	3.2686876
1	2012	3.1962251
1	2013	2.8720914
1	2014	1.4866131
1	2015	3.5742382
1	2016	3.6273793
1	2017	2.3793960

1-10 of 113 rows   Previous 1 2 3 4 5 6 ... 12 Next

```
weather.rain |>
  ggplot() +
  geom_line(aes(x = Month, y = RainAvg, color = "rainfall")) +
  facet_wrap(~Year, ncol = 4) +
  ggtitle("Average Rainfall in Australia over Time") +
  labs(y = "Average Rainfall", color = "Legend") +
  scale_color_manual(values = c("rainfall" = "deepskyblue")) +
  scale_x_continuous(breaks = 1:12, labels = month.name) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  + # Tilt month names by 45 d
  theme(legend.position = "none")
```



7. Comment on your faceted visualization.

This graph shows the average rainfall per month in Australia from 2007 to 2017. Again, this graph is very simple to read, which is its biggest strength. Each year is clearly labelled, and it makes for easy comparisons. For example, 2014 had very little rain, while 2008 had a huge amount of rain in February. Again, because there are so many grids and months, I wish the text for the months was bigger and easier to read. Otherwise, this is a decent figure.