Chapter 2 Notes

2.1 tsibble objects

Formerly a `ts()` object, `tsibble()` is the new way to establish a timeseries object

```
y <- tsibble(
  Year = 2015:2019,
  Observation = c(123, 39, 78, 52, 110),
  index = Year
)
```

`tsibble` objects extend tidy data frames (`tibble` objects) by introducing temporal structure.

When observations are more frequent that yearly, a timeclass function must be used as the index. Below is a monthly `tibble` df:

```
#> # A tibble: 5 x 2
#>   Month    Observation
#>   <chr>          <dbl>
#> 1 2019 Jan          50
#> 2 2019 Feb          23
#> 3 2019 Mar          34
#> 4 2019 Apr          30
#> 5 2019 May          25
```

In order to convert to `tsibble`, convert the `Month` column from `<chr>` to `<mth>` using `yearmonth()` and identifying `index` variable with `as_tsibble()`:

```
z %>%
  mutate(Month = yearmonth(Month)) %>%
  as_tsibble(index = Month)
```

Other Time Class Functions

Feature	Function
Annual	`start:end`
Quarterly	`yearquarter()`
Monthly	`yearmonth()`
Weekly	`yearweek()`
Daily	`as_date()`, `ymd()`
Sub-daily	`as_datetime()`, `ymd_hms()`

Working with `tsibble` Objects

We can use `dplyr` functions on `tsibble` objects.Examples below using the PBS tsibble containing sales data on pharmaceutical products in Australia

```
PBS
```

```
## # A tsibble: 67,596 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [336]
```

```
##       Month Concession  Type   ATC1  ATC1_desc   ATC2  ATC2_desc   Scripts  Cost
##       <mth> <chr>       <chr>  <chr> <chr>       <chr> <chr>         <dbl> <dbl>
##  1 1991 Jul Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   18228 67877
##  2 1991 Aug Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15327 57011
##  3 1991 Sep Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   14775 55020
##  4 1991 Oct Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15380 57222
##  5 1991 Nov Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   14371 52120
##  6 1991 Dec Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15028 54299
##  7 1992 Jan Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   11040 39753
##  8 1992 Feb Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   15165 54405
##  9 1992 Mar Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   16898 61108
## 10 1992 Apr Concession~ Co-pa~ A     Alimentary~ A01   STOMATOLOG~   18141 65356
## # ... with 67,586 more rows
```

Using the `filter()` function to call specific value from column:

```
PBS %>%
  filter(ATC2 == 'A10')
```

```
## # A tsibble: 816 x 9 [1M]
## # Key:       Concession, Type, ATC1, ATC2 [4]
##       Month Concession  Type   ATC1  ATC1_desc   ATC2  ATC2_desc  Scripts   Cost
##       <mth> <chr>       <chr>  <chr> <chr>       <chr> <chr>        <dbl>  <dbl>
##  1 1991 Jul Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   89733 2.09e6
##  2 1991 Aug Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   77101 1.80e6
##  3 1991 Sep Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   76255 1.78e6
##  4 1991 Oct Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   78681 1.85e6
##  5 1991 Nov Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   70554 1.69e6
##  6 1991 Dec Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   75814 1.84e6
##  7 1992 Jan Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   64186 1.56e6
##  8 1992 Feb Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   75899 1.73e6
##  9 1992 Mar Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   89445 2.05e6
## 10 1992 Apr Concession~ Co-pa~ A     Alimentary~ A10   ANTIDIABE~   97315 2.23e6
## # ... with 806 more rows
```

Selecting the specific columns we need with `select()`:

```
PBS %>%
  filter(ATC2 == 'A10') %>%
  select(Month, Concession, Type, Cost)
```

```
## # A tsibble: 816 x 4 [1M]
## # Key:       Concession, Type [4]
##       Month Concession   Type           Cost
##       <mth> <chr>        <chr>         <dbl>
##  1 1991 Jul Concessional Co-payments 2092878
##  2 1991 Aug Concessional Co-payments 1795733
##  3 1991 Sep Concessional Co-payments 1777231
##  4 1991 Oct Concessional Co-payments 1848507
##  5 1991 Nov Concessional Co-payments 1686458
##  6 1991 Dec Concessional Co-payments 1843079
##  7 1992 Jan Concessional Co-payments 1564702
##  8 1992 Feb Concessional Co-payments 1732508
##  9 1992 Mar Concessional Co-payments 2046102
## 10 1992 Apr Concessional Co-payments 2225977
## # ... with 806 more rows
```

`select()` handles columns while `filter()` handles rows

`summarize()` allows you to combine data across keys:

```
PBS %>%
  filter(ATC2 == 'A10') %>%
  select(Month, Concession, Type, Cost) %>%
  summarize(TotalC = sum(Cost))
```

```
## # A tsibble: 204 x 2 [1M]
##       Month  TotalC
##       <mth>   <dbl>
##  1 1991 Jul 3526591
##  2 1991 Aug 3180891
##  3 1991 Sep 3252221
##  4 1991 Oct 3611003
##  5 1991 Nov 3565869
##  6 1991 Dec 4306371
##  7 1992 Jan 5088335
##  8 1992 Feb 2814520
##  9 1992 Mar 2985811
## 10 1992 Apr 3204780
## # ... with 194 more rows
```

Creating new variables using `mutate()`

```
PBS %>%
  filter(ATC2 == 'A10') %>%
  select(Month, Concession, Type, Cost) %>%
  summarize(TotalC = sum(Cost)) %>%
  mutate(Cost = TotalC/1e6)
```

```
## # A tsibble: 204 x 3 [1M]
##       Month  TotalC  Cost
##       <mth>   <dbl> <dbl>
##  1 1991 Jul 3526591  3.53
##  2 1991 Aug 3180891  3.18
##  3 1991 Sep 3252221  3.25
##  4 1991 Oct 3611003  3.61
##  5 1991 Nov 3565869  3.57
##  6 1991 Dec 4306371  4.31
##  7 1992 Jan 5088335  5.09
##  8 1992 Feb 2814520  2.81
##  9 1992 Mar 2985811  2.99
## 10 1992 Apr 3204780  3.20
## # ... with 194 more rows
```

Saving as a tsibble():

```
a10 <- PBS %>%
  filter(ATC2 == 'A10') %>%
  select(Month, Concession, Type, Cost) %>%
  summarize(TotalC = sum(Cost)) %>%
  mutate(Cost = TotalC/1e6)
```

Reading CSVs

```
prison <- readr::read_csv("https://OTexts.com/fpp3/extrafiles/prison_population.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Date = col_date(format = ""),
##   State = col_character(),
##   Gender = col_character(),
##   Legal = col_character(),
##   Indigenous = col_character(),
##   Count = col_double()
## )
```

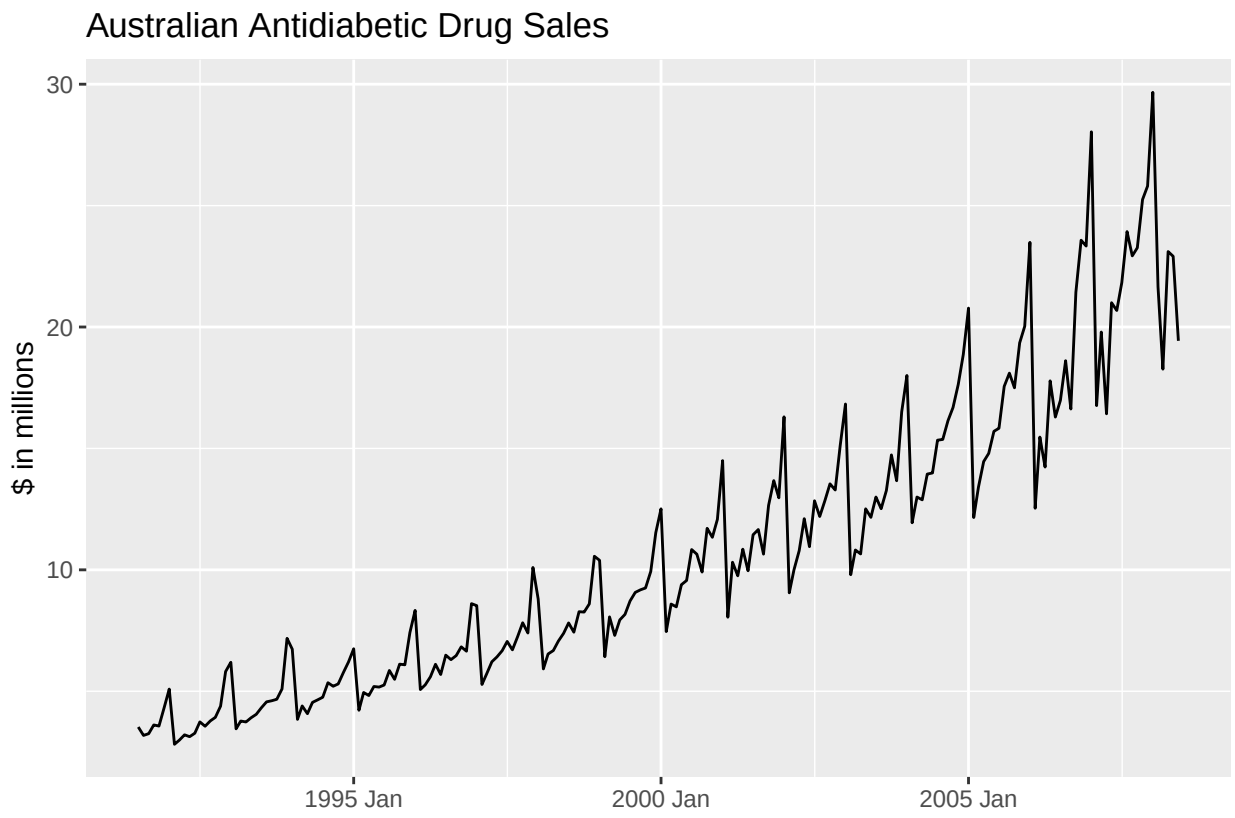
```
# The original CSV has the date variable as individual days and they should be quarters

prison <- prison %>%
  mutate(Quarter = yearquarter(Date)) %>%
  select(-Date) %>%
  as_tsibble(key = c(State, Gender, Legal, Indigenous),
             index = Quarter)

prison
```

```
## # A tsibble: 3,072 x 6 [1Q]
## # Key:       State, Gender, Legal, Indigenous [64]
##    State Gender Legal    Indigenous Count Quarter
##    <chr> <chr>  <chr>    <chr>      <dbl>   <qtr>
##  1 ACT   Female Remanded ATSI           0 2005 Q1
##  2 ACT   Female Remanded ATSI           1 2005 Q2
##  3 ACT   Female Remanded ATSI           0 2005 Q3
##  4 ACT   Female Remanded ATSI           0 2005 Q4
##  5 ACT   Female Remanded ATSI           1 2006 Q1
##  6 ACT   Female Remanded ATSI           1 2006 Q2
##  7 ACT   Female Remanded ATSI           1 2006 Q3
##  8 ACT   Female Remanded ATSI           0 2006 Q4
##  9 ACT   Female Remanded ATSI           0 2007 Q1
## 10 ACT   Female Remanded ATSI           1 2007 Q2
## # ... with 3,062 more rows
```

2.2 Time Plots

Example of a time plot

```
autoplot(a10, Cost) +
  labs(y='$ in millions',
       x='',
      title= 'Australian Antidiabetic Drug Sales')
```



Plot shows:

- clear and increasing trend
- strong seasonal pattern
- increase in variance

Reason behind the shape:

- Government subsidizes in such a way that makes it cost-effective for patients to stockpile at the end of the calendar year, which leads to the drop at the beginning of each year.

2.3 Time Series Patterns

Trend: A long-term increase or decrease in the data.

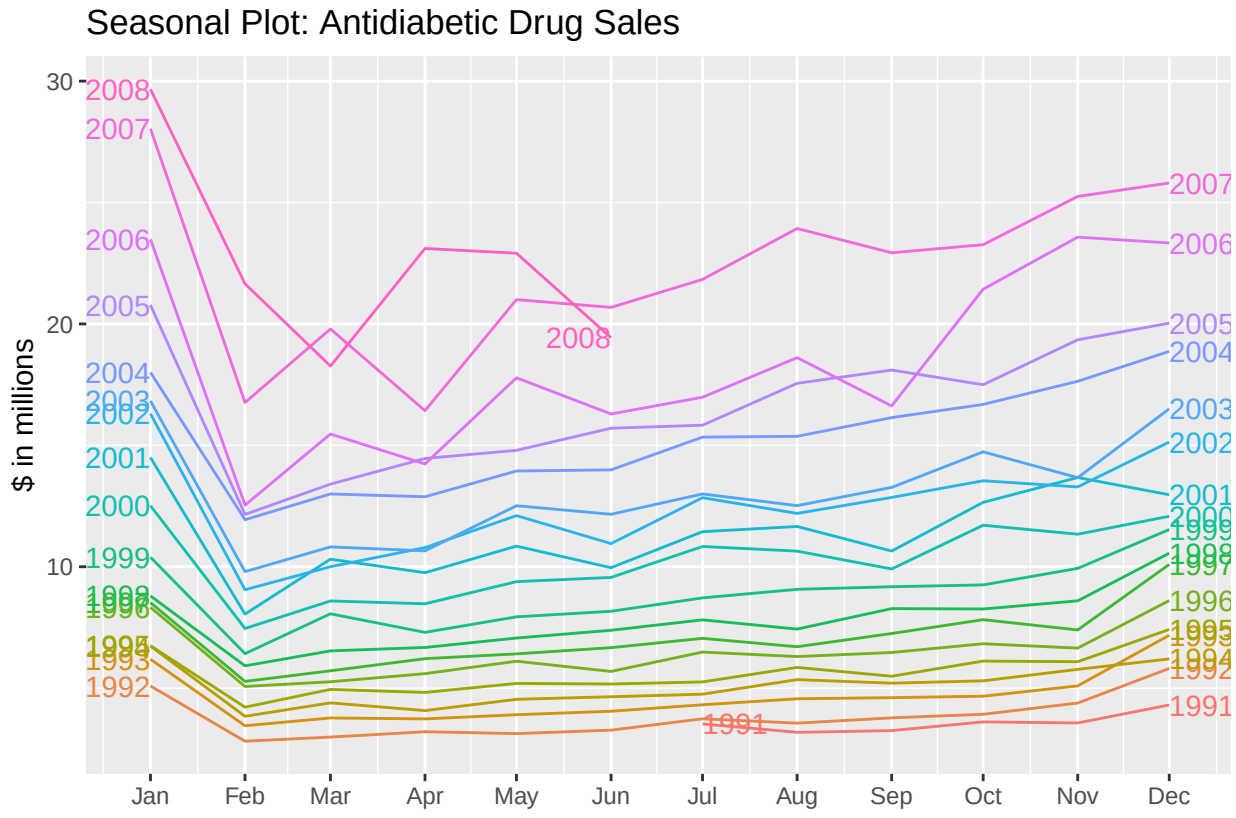
Seasonality: When a pattern occurs in a fixed and known period of time, typically a yearly or less (i.e. hourly, weekly, monthly, quarterly).

Cycle: Occurs when the data exhibits a pattern not of fixed frequency usually on a scale $>$ 2 years (i.e. market downturns every 7-10 years)

2.4 Seasonal Plots

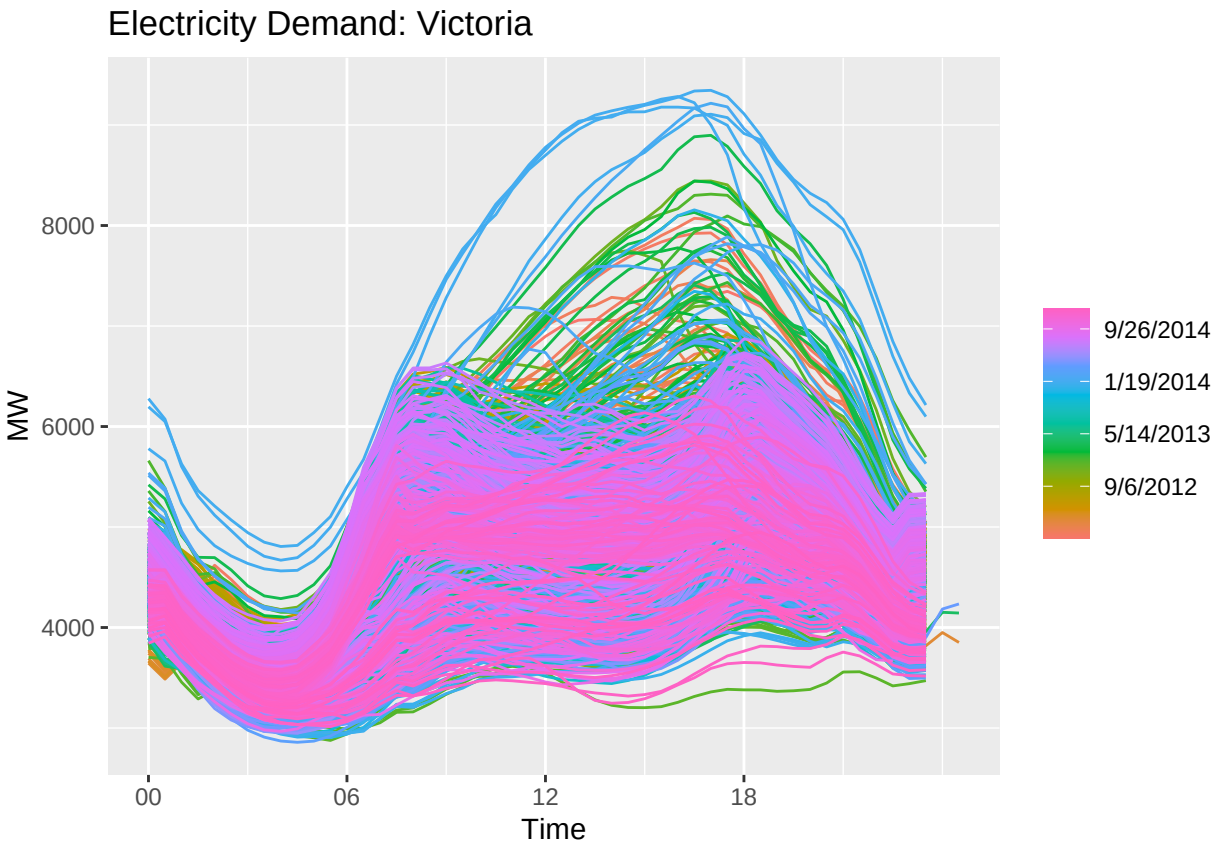
A seasonal plot shows the data plotted against each individual "season"

```
a10 %>%
  gg_season(Cost, labels='both') +
  labs(y = '$ in millions',
       x='',
       title = 'Seasonal Plot: Antidiabetic Drug Sales') +
  expand_limits(x = ymd(c('1972-12-28','1973-12-04')))
```

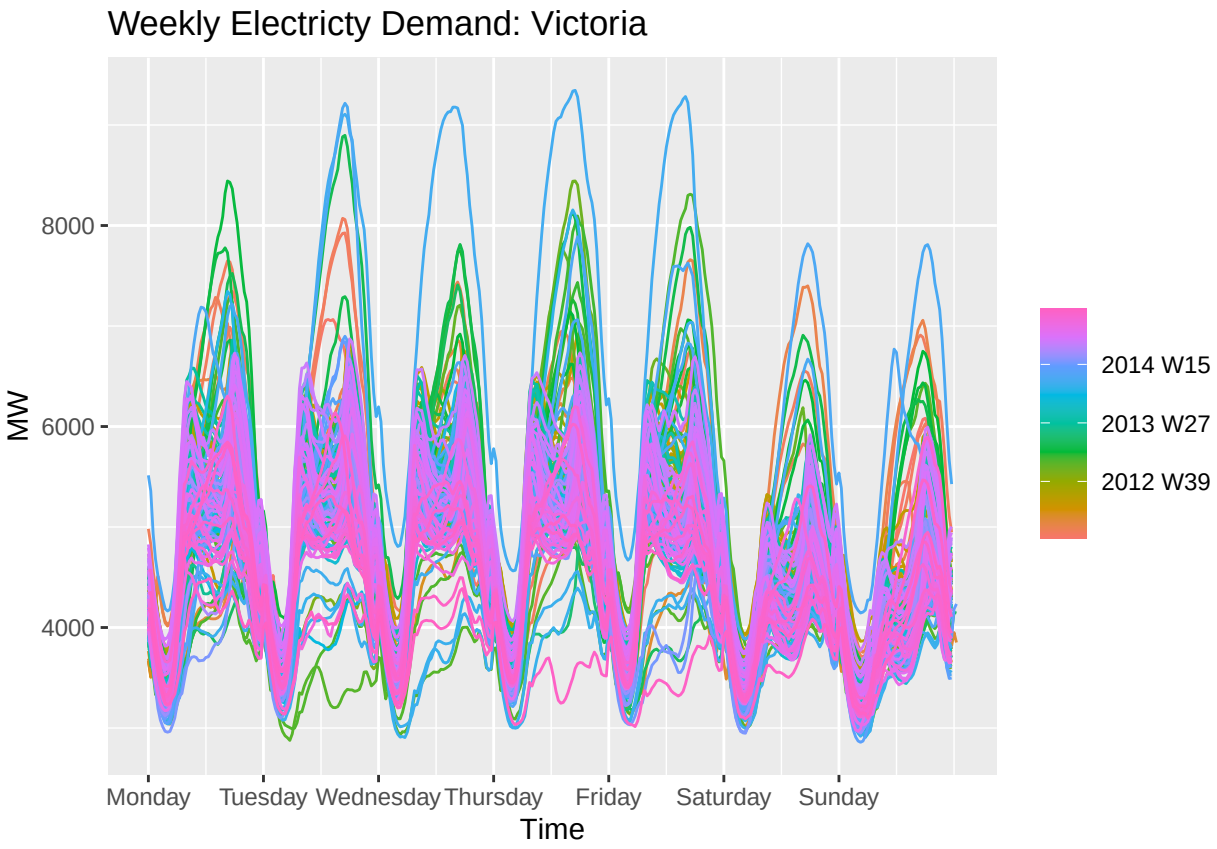


Multiple Seasonal Periods In a case where data has more than one season pattern, use `period` argument.

```
vic_elec %>% gg_season(Demand, period='day') +
  #theme(legend.position = 'none') + #very unclear without the legend
  labs(y='MW', title='Electricity Demand: Victoria')
```



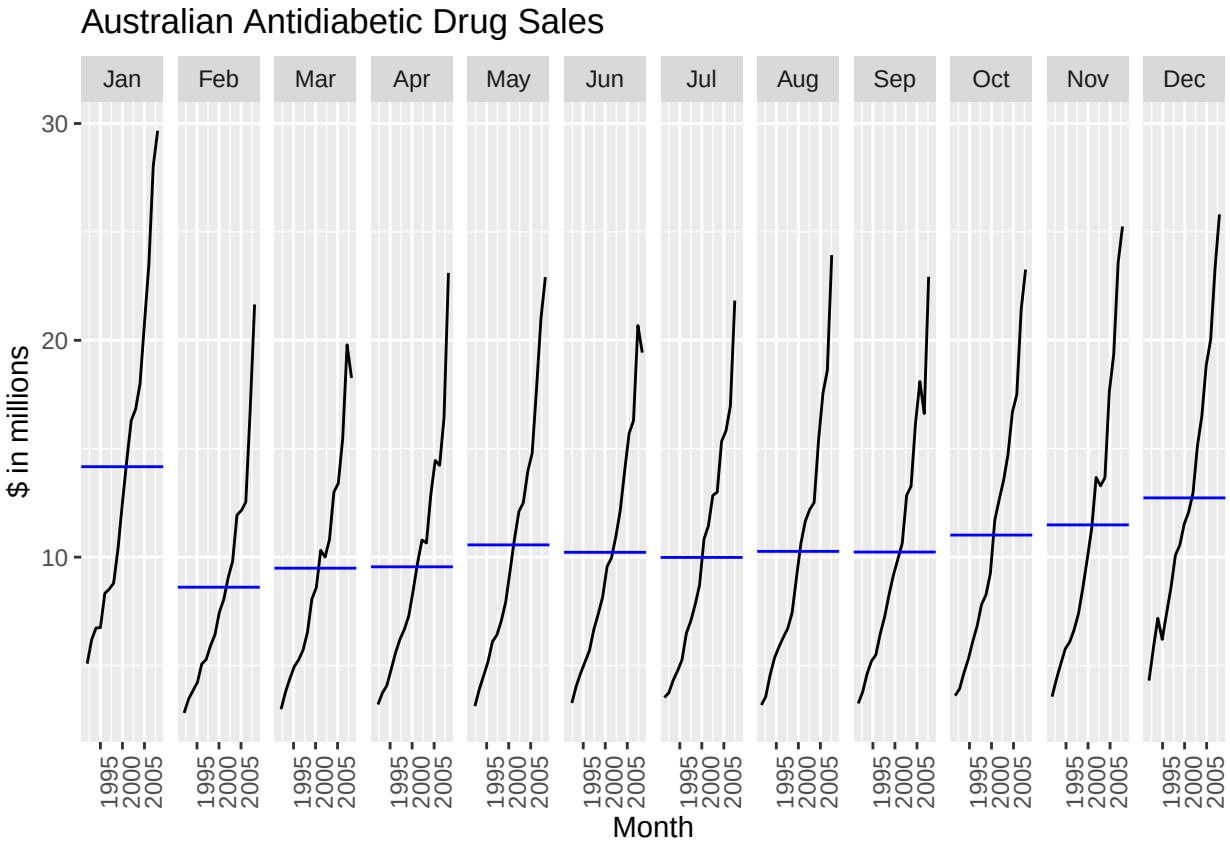

```
vic_elec %>% gg_season(Demand, period='week') +
  labs(y='MW',title='Weekly Electricty Demand: Victoria')
```



2.5 Seasonal Subseries Plot

Alternative plot where data from each season is collected in a mini time plot

```
a10 %>%
  gg_subseries(Cost)+
  labs(y = '$ in millions',
       title = 'Australian Antidiabetic Drug Sales'
  )
```



What the plot shows:

- Blue horizontal lines indicate means for each month.
- Seasonal pattern can be seen clearly
- Shows seasonal changes over time

2.6