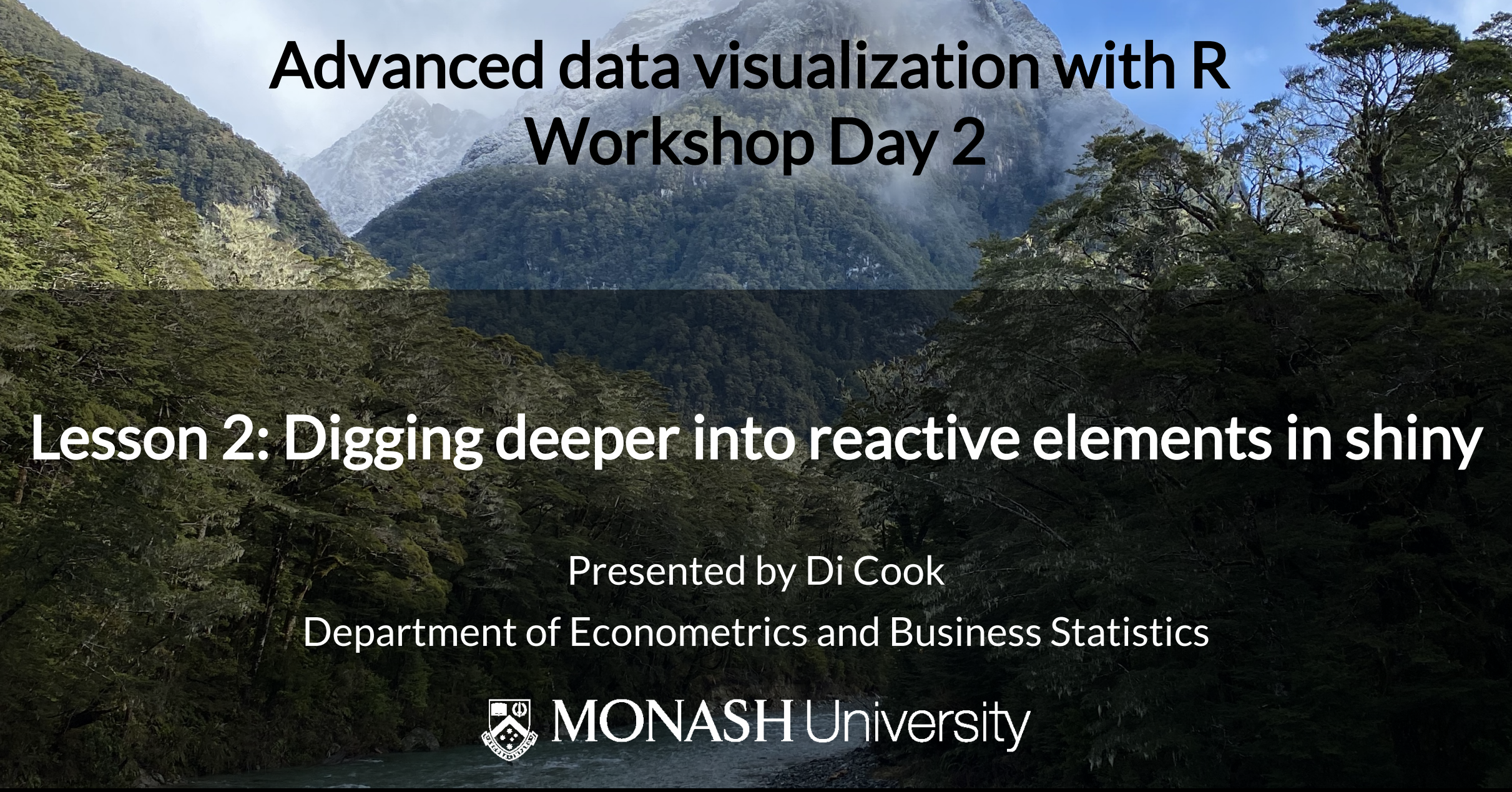


Advanced data visualization with R
Workshop Day 2

Lesson 2: Digging deeper into reactive elements in shiny

Presented by Di Cook

Department of Econometrics and Business Statistics

MONASH University

8-9th Dec 2021 @ Statistical Society of Australia | Zoom

Outline

Animation
Linking between plots
Using a `key` variable to link between plots

Animation

Animation allows for controlled changes in a plot. It can provide more information to be displayed than is possible within a single plot. It can also draw attention to specific parts of the plot.

Human perception is fleeting, though, and prone to forgetting what was just displayed. Keeping some elements persistent, possibly faint, or light coloured, can be useful for perceiving patterns of change.

Think about what part of a plot would lend itself to animation, eg a time variable, or possibly a categorical variable.

Primary tools: `plotly`, `gganimate`

```
# Schema for animation
ggplot %>%
  animation %>%
  options (transition,
           speed,
           fade)
```

demo data for animation

ABS and Election data

Show 10 entries Search:

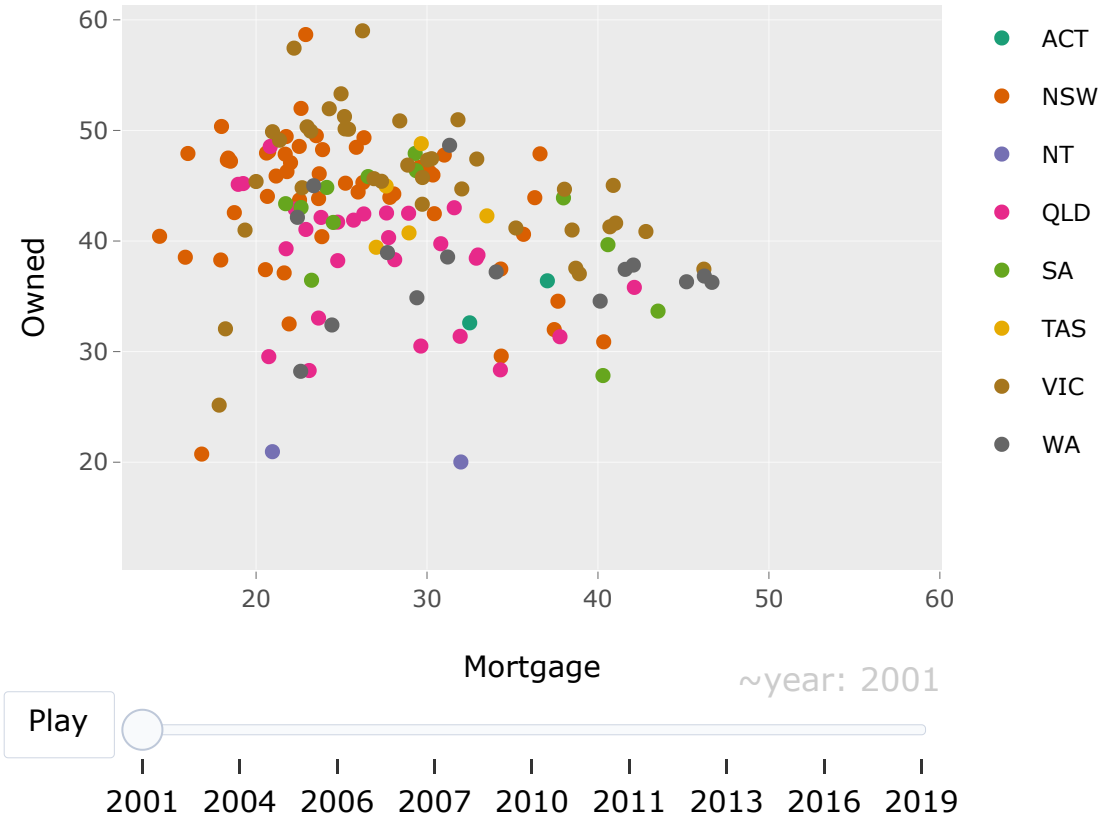
	year	UniqueID	State	Mortgage	Owned	Renting	Unemployed
1	2001	401	SA	23.24	36.45	37.22	7.78
2	2001	201	VIC	40.89	45.03	11.98	4.72
3	2001	202	VIC	29.73	45.74	22.21	8.96
4	2001	101	NSW	22.53	48.56	26.83	5.30
5	2001	402	SA	26.55	45.83	22.94	6.33
6	2001	102	NSW	17.98	50.36	29.12	5.80
7	2001	601	TAS	28.95	40.73	27.84	9.66
8	2001	203	VIC	20.00	45.38	32.37	8.92
9	2001	204	VIC	28.87	46.87	22.04	9.11
10	2001	103	NSW	21.17	45.88	30.32	4.49

Showing 1 to 10 of 1,316 entries

Previous 1 2 3 4 5 ... 132 Next

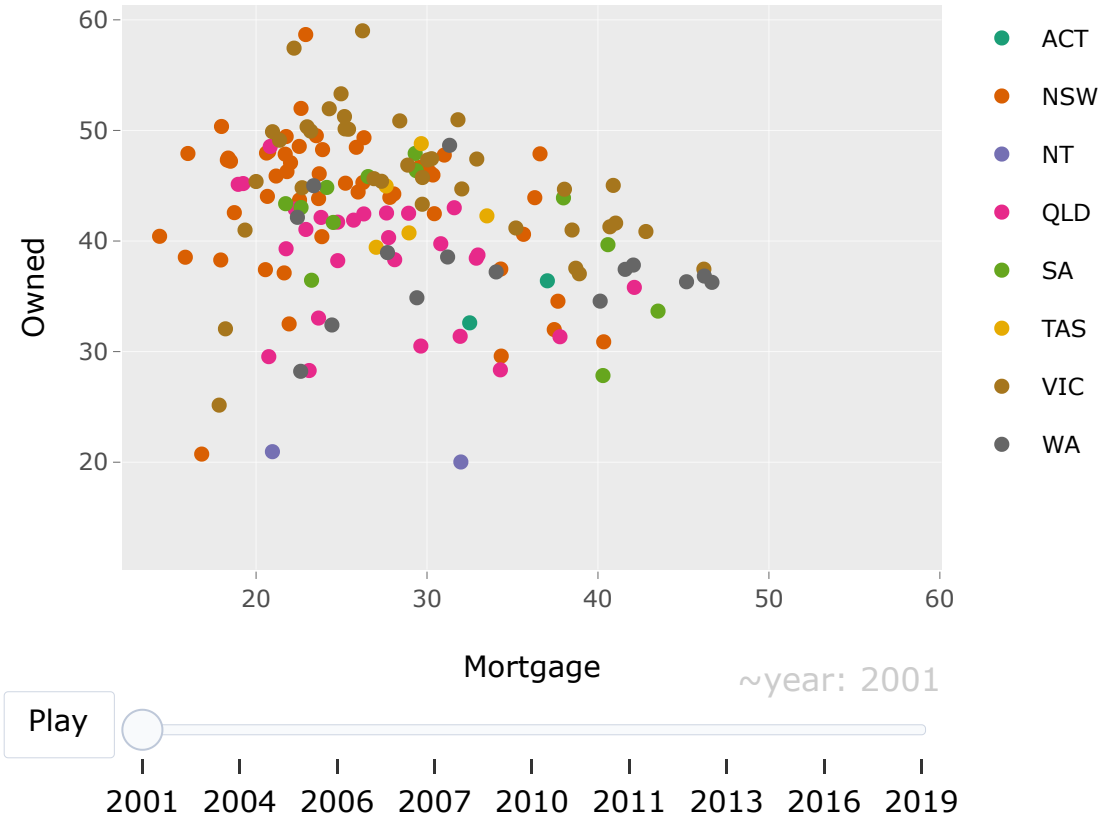
Animation with plotly

```
abs_p <- ggplot(abs, aes(
    x=Mortgage,
    y=Owned,
    colour=State)) +
  geom_point(aes(frame=year,
                 ids=UniqueID)) +
  scale_colour_brewer("",
     palette = "Dark2")
ggplotly(abs_p)
```



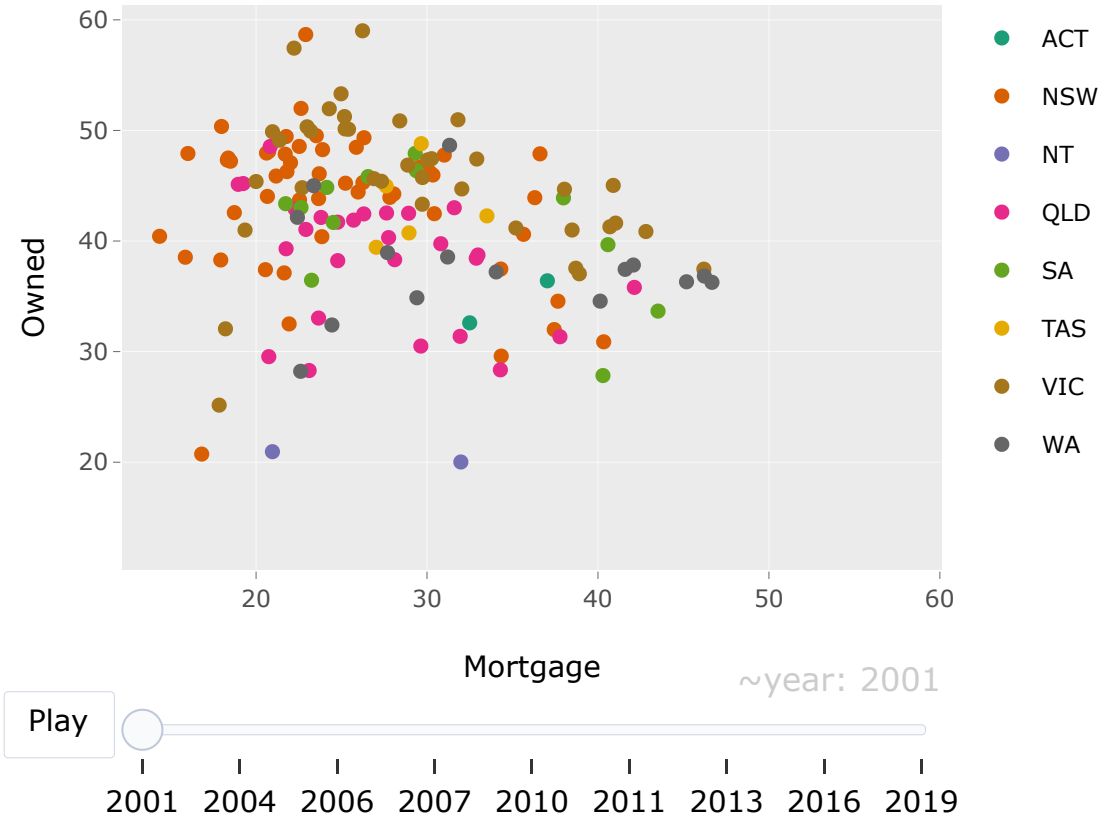
Animation with plotly

```
abs_p <- ggplot(abs, aes(
    x=Mortgage,
    y=Owned,
    colour=State)) +
  geom_point(aes(frame=year,
                 ids=UniqueID)) +
  scale_colour_brewer("",
     palette = "Dark2")
ggplotly(abs_p) %>%
  animation_opts(transition=0)
```



Animation with plotly

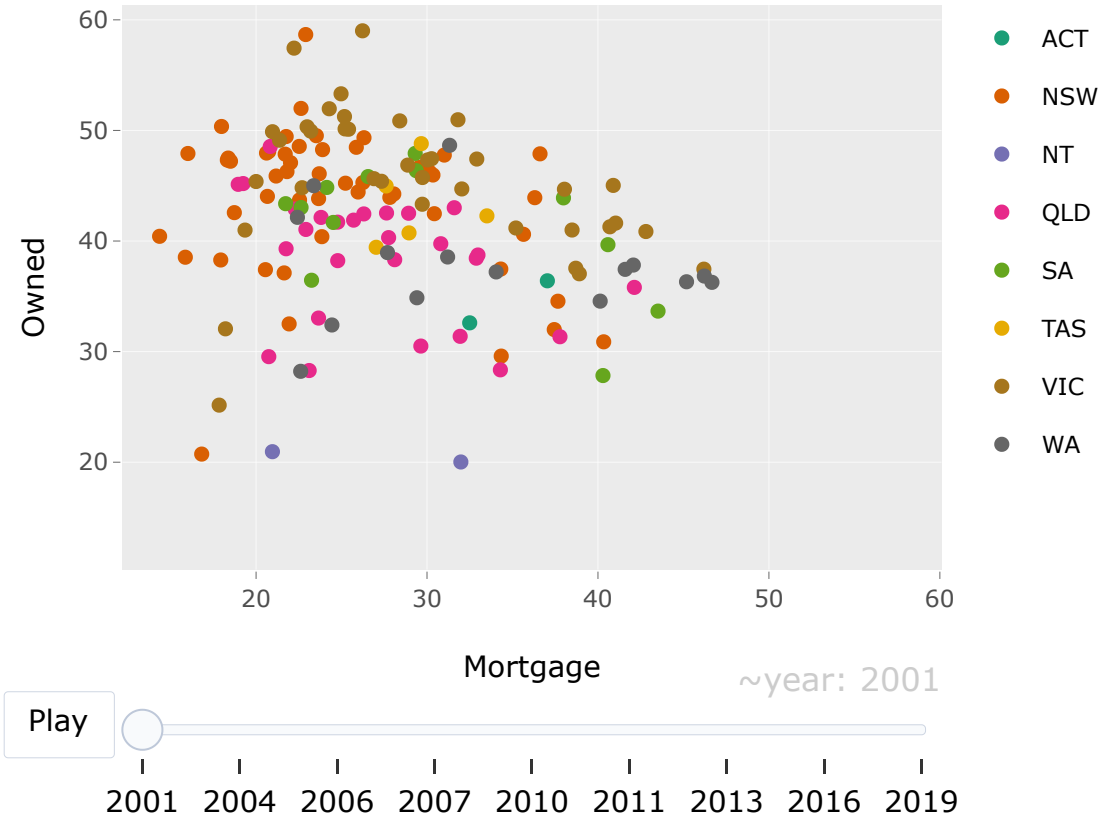
```
abs_p <- ggplot(abs, aes(
    x=Mortgage,
    y=Owned,
    colour=State)) +
  geom_point(aes(frame=year,
                 ids=UniqueID)) +
  scale_colour_brewer("",
     palette = "Dark2")
ggplotly(abs_p) %>%
  animation_opts(frame=1000,
                 transition=950,
                 easing="cubic")
```



Animation with plotly

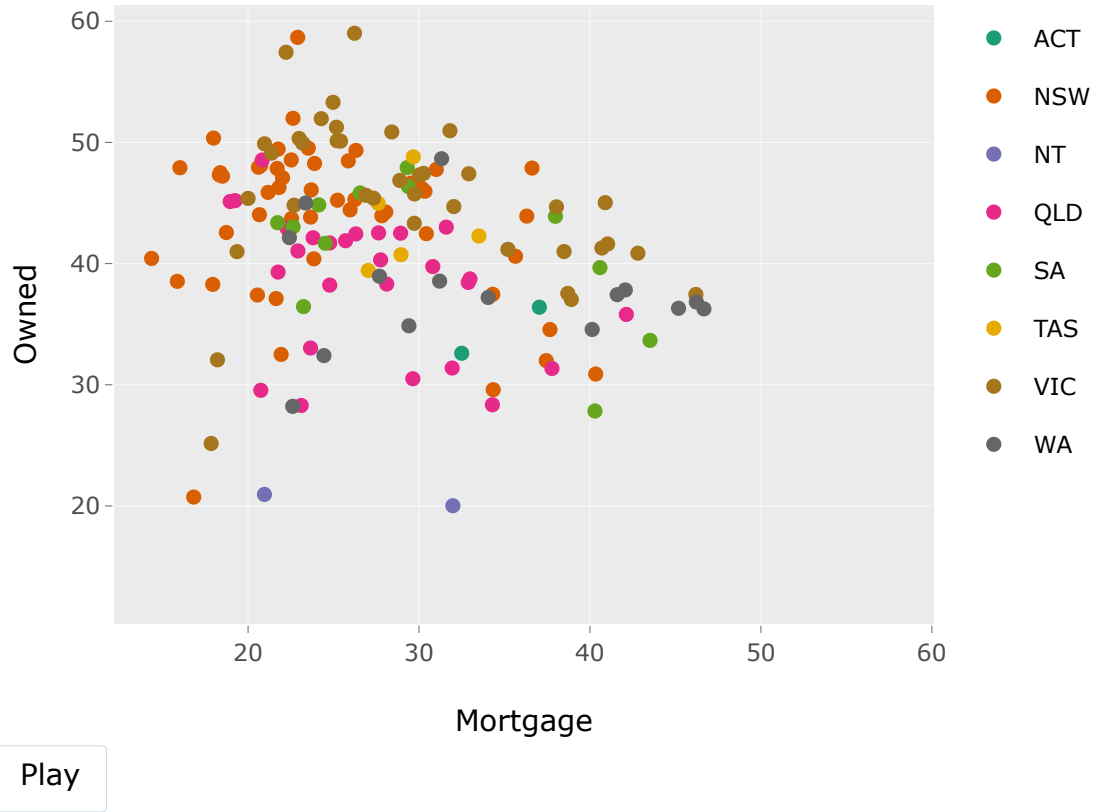
```
abs_p2 <- ggplot(abs, aes(
    x=Mortgage,
    y=Owned,
    colour=State)) +
  geom_point(aes(frame=year)) +
  scale_colour_brewer("",
     palette = "Dark2")
ggplotly(abs_p2) %>%
  animation_opts(frame=1000,
                 transition=950)
```

The `ids` match up the points from one frame to another. If they aren't set the transition can look chaotic.



Animation with plotly

```
abs_p <- ggplot(abs, aes(
    x=Mortgage,
    y=Owned,
    colour=State)) +
  geom_point(aes(frame=year,
                 ids=UniqueID)) +
  scale_colour_brewer("",
     palette = "Dark2")
ggplotly(abs_p) %>%
  animation_slider(hide=TRUE)
```

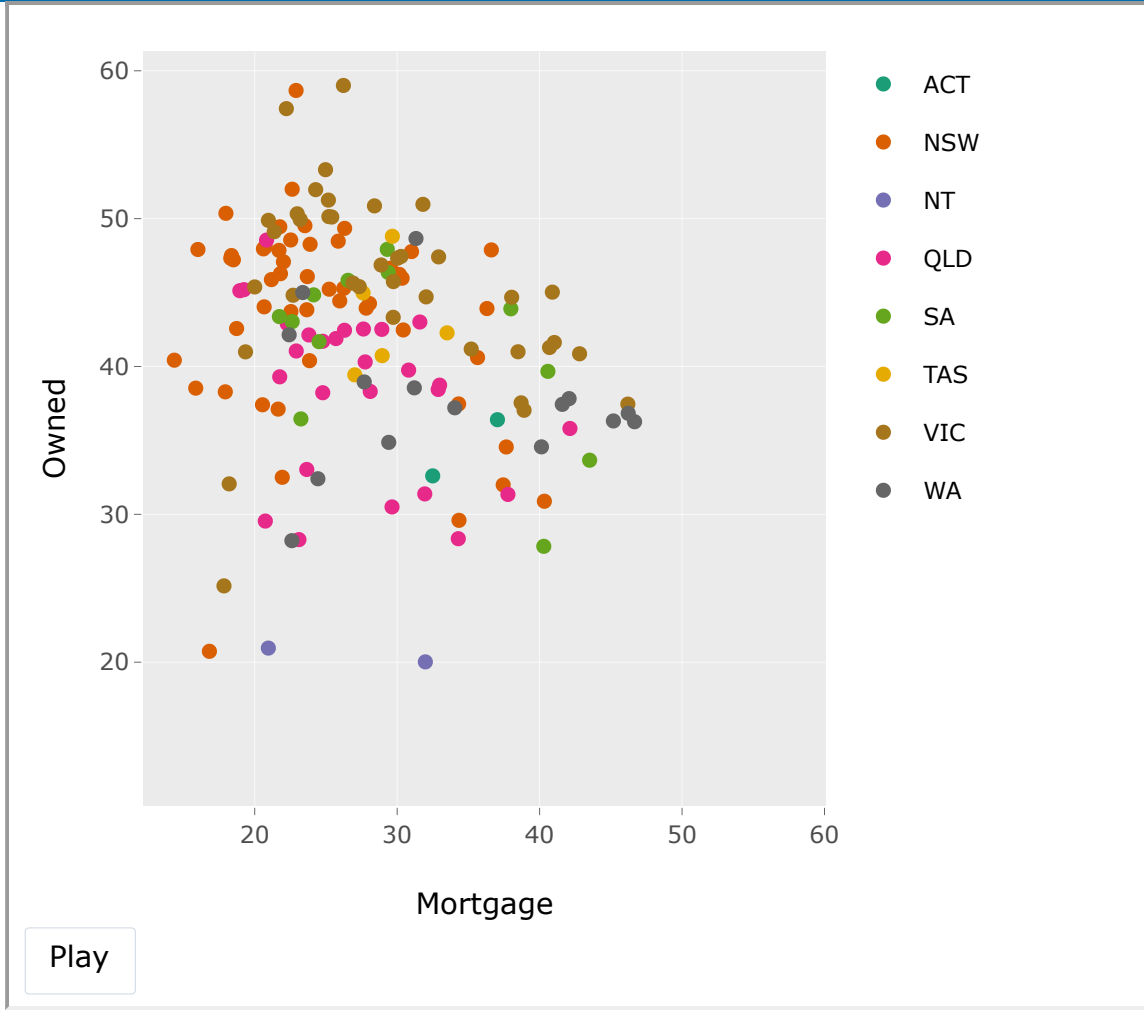


Saving your animation

```
library(htmlwidgets)
abs_ply <- ggplotly(abs_p,
                    width=500,
                    height=500) %>%
  animation_slider(hide=TRUE)
saveWidget(abs_ply, "abs_ply.html",
           selfcontained = F)
```

To embed it in the html document:

```
<iframe
src="images/abs_ply.html"
width="100%" height="500"
</iframe>
```



Animation with gganimate

The `gganimate` package is recommended for more control of an animation. There are more choices for different transitions, easing, traces from previous position.

Construction follows the same as for `plotly`: generate your ggplot, specify the animation, add options.

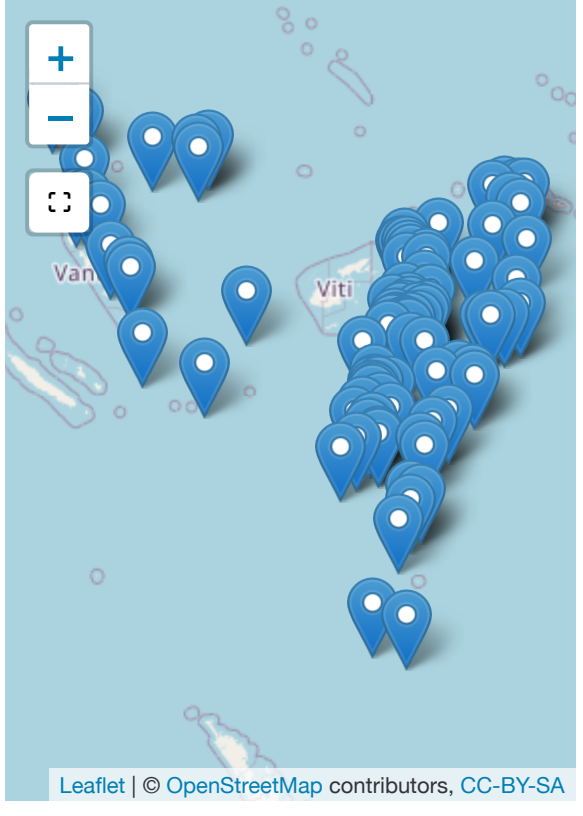
Linking between plots

Using crosstalk to link information between plots

Crosstalk extends htmlwidgets with a set of classes, functions, and conventions for implementing cross-widget interactions (currently, linked brushing and filtering).

```
# Wrap data frame in SharedData
sd <- SharedData$new(
  quakes[sample(nrow(quakes),
                100),])

# Use SharedData like a dataframe wit
bscols(
  leaflet(sd) %>%
    addTiles() %>%
    addMarkers(),
  datatable(sd, class="compact",
            width="200"))
```



Show 10 entries

Search:

	lat	long	depth	ma
669	-23.82	180.09	498	
80	-28.98	181.11	304	
847	-25.63	180.26	464	
953	-18.68	184.5	174	
337	-25.8	182.1	68	
809	-20.9	182.02	402	
882	-21.2	181.4	560	
704	-25	180	488	
278	-24.78	179.22	492	
598	-17.02	182.93	406	

Showing 1 to 10 of 100 entries

Mapping out the data analysis: ABS

Show 10 entries Search:

	year	UniqueID	State	Mortgage	Owned	Renting	Unemployed
1	2001	401	SA	23.24	36.45	37.22	7.78
2	2001	201	VIC	40.89	45.03	11.98	4.72
3	2001	202	VIC	29.73	45.74	22.21	8.96
4	2001	101	NSW	22.53	48.56	26.83	5.30
5	2001	402	SA	26.55	45.83	22.94	6.33
6	2001	102	NSW	17.98	50.36	29.12	5.80
7	2001	601	TAS	28.95	40.73	27.84	9.66
8	2001	203	VIC	20.00	45.38	32.37	8.92
9	2001	204	VIC	28.87	46.87	22.04	9.11
10	2001	103	NSW	21.17	45.88	30.32	4.49

Showing 1 to 10 of 1,316 entries

Previous 1 2 3 4 5 ... 132 Next

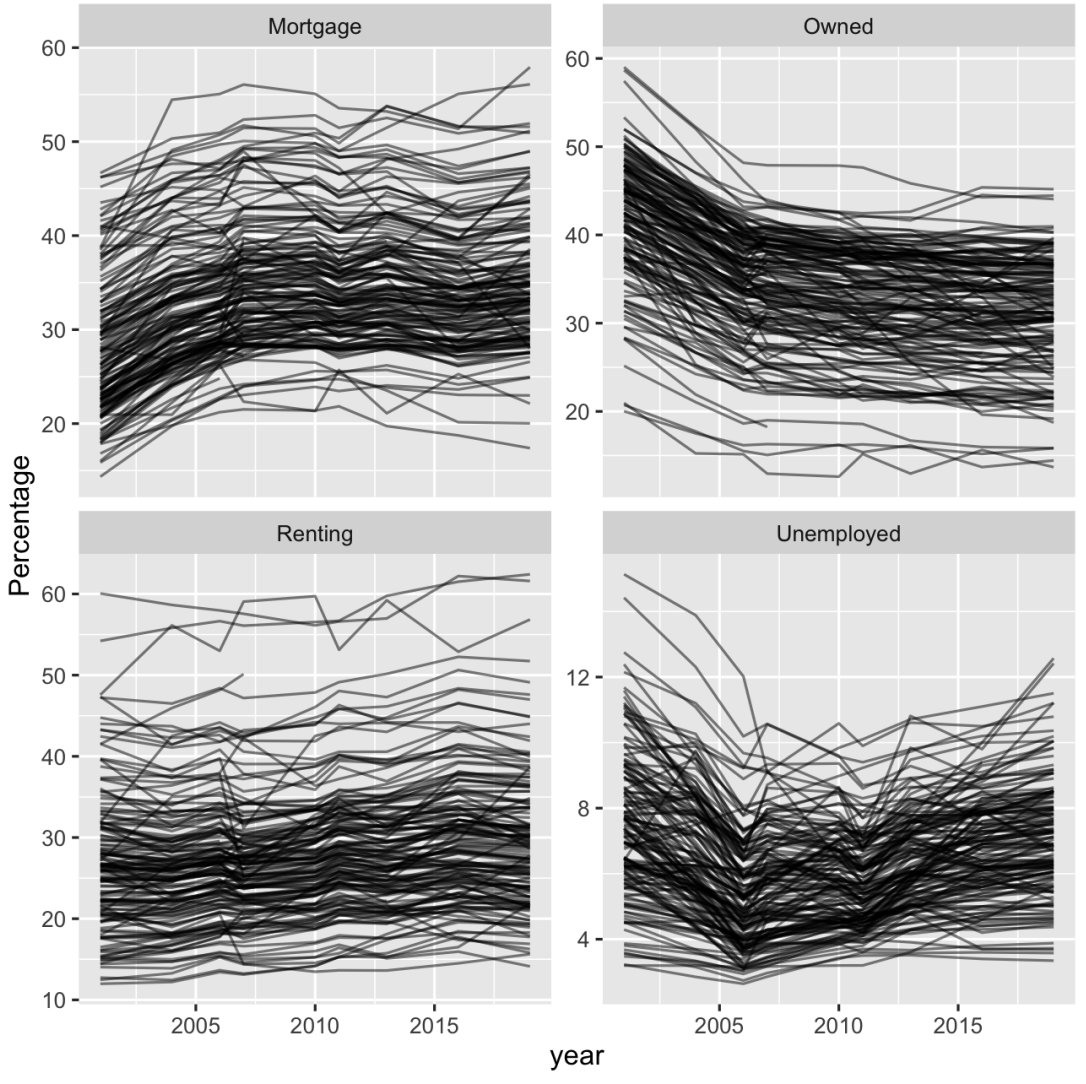
Mapping out the data analysis: ABS

Think about the questions that might be answered with the data

- Is the percentage of home ownership changing over the time period?
- What is the relationship between home ownership and having a mortgage?
- Is home ownership lower where there is higher unemployment?
- Is there a difference across states in percentage of renters?

Trends over time

```
abs_long <- abs %>%
  pivot_longer("Mortgage":"Unemployed",
               names_to ="Statistic",
               values_to = "Percentage")
ggplot(abs_long, aes(x=year,
                     y=Percentage)) +
  geom_line(aes(group=UniqueID),
            alpha=0.5) +
  facet_wrap(~Statistic,
             scales="free_y")
```



Trends over time and linking

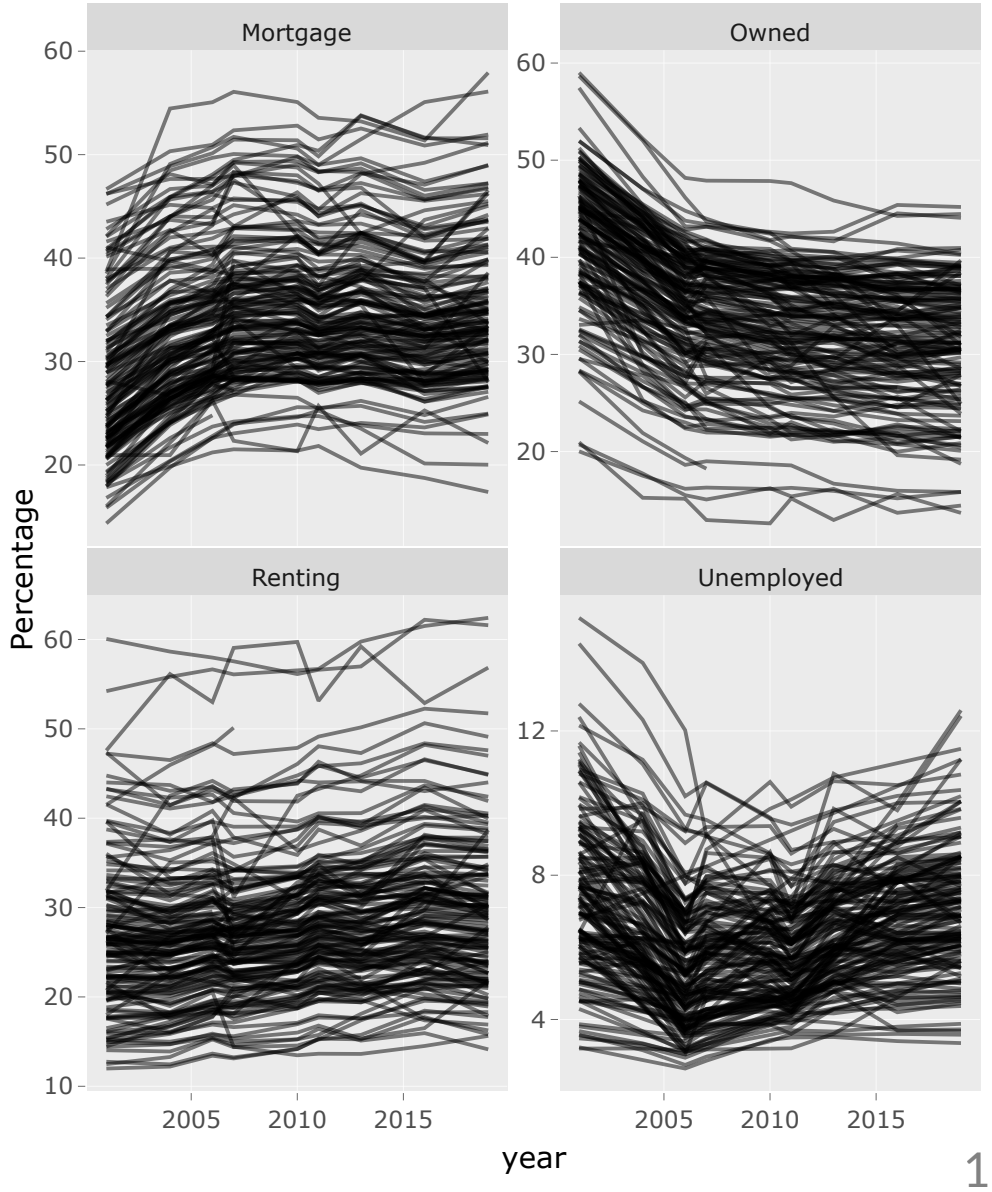
```
abs_key <- highlight_key(abs_long,
                         ~UniqueID)

abs_p2 <- ggplot(abs_key,
                 aes(x=year,
                     y=Percentage)) +
  geom_line(aes(group=UniqueID),
            alpha=0.5) +
  facet_wrap(~Statistic,
             scales="free_y")

abs_gg2 <- ggplotly(abs_p2,
                    height = 600,
                    width = 500)

highlight(abs_gg2)
```

Click on a line



Linking control

`highlight_key()` is an implementation of `crosstalk::SharedData()`

`highlight()` controls the options for selection.

Trends over time and linking

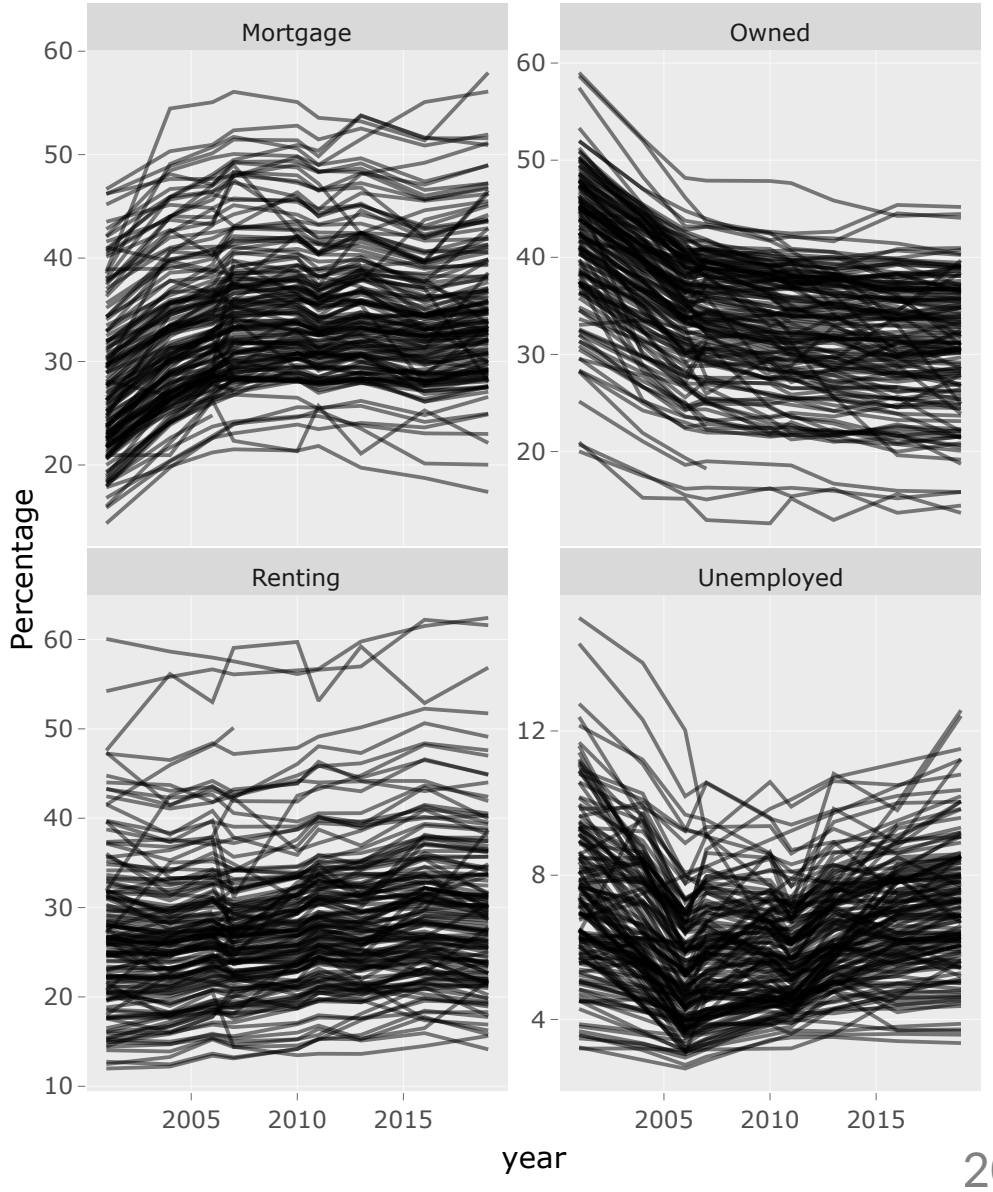
```
abs_key2<- highlight_key(abs_long,
                         ~State)

abs_p3 <- ggplot(abs_key2,
                 aes(x=year,
                     y=Percentage,
                     text=State)) +
  geom_line(aes(group=UniqueID),
            alpha=0.5) +
  facet_wrap(~Statistic,
             scales="free_y")

abs_gg3 <- ggplotly(abs_p3,
                    height = 600,
                    width = 500)

highlight(abs_gg3)
```

Click on a line



Exploration vs explanation

These two displays were in support of exploration.

We didn't know a lot about the data.

Based on the data description we identified several questions of interest.

The linked plots assisted in answering the question about temporal trend, and also relative to state.

demo

Special purpose linked plots

```
library(tsibble)
library(tsibbletalk)
## ---- tourism-shared
tourism_shared <- tourism_monthly %>%
  as_shared_tsibble(
    spec = (State / Region) * Purpose)
## ---- plotly-key-tree
p_l <- plotly_key_tree(tourism_shared,
                       height = 1100,
                       width = 800)
## ---- tourism-series
p_tr <- tourism_shared %>%
  ggplot(aes(x = Month, y = Trips)) +
  geom_line(aes(group = Region), alpha = .5, size = .4) +
  facet_wrap(~ Purpose, scales = "free_y") +
  scale_x_yearmonth(date_breaks = "5 years", date_labels = "%Y")
## ---- tourism-scatter
library(feasts)
```

Open `day2-exercise-02.Rmd`

15:00

Learning more

- Sievert (2019) Interactive web-based data visualization with R, plotly, and shiny
- RStudio's htmlwidgets gallery
- RStudio's crosstalk guide
- tsibbletalk

Session Information

```
devtools::session_info()
```

```
## ─ Session info ─────────────────────────────────────────────────────────────
##  setting  value
##  version  R version 4.1.0 (2021-05-18)
##  os       macOS Big Sur 10.16
##  system   x86_64, darwin17.0
##  ui       X11
##  language (EN)
##  collate  en_AU.UTF-8
##  ctype    en_AU.UTF-8
##  tz       Australia/Melbourne
##  date     2021-12-09
##
## ─ Packages ─────────────────────────────────────────────────────────────────
##  package        * version     date       lib
##  anicon           0.1.0       2021-07-14 [1]
##  anytime          0.3.9       2020-08-27 [1]
##  assertthat       0.2.1       2019-03-21 [1]
##  backports        1.2.1       2020-12-09 [1]
##  broom            0.7.9       2021-07-27 [1]
##  bslib            0.3.1       2021-10-06 [1]
```