ETC5510: Introduction to Data Analysis

Week 5, part A

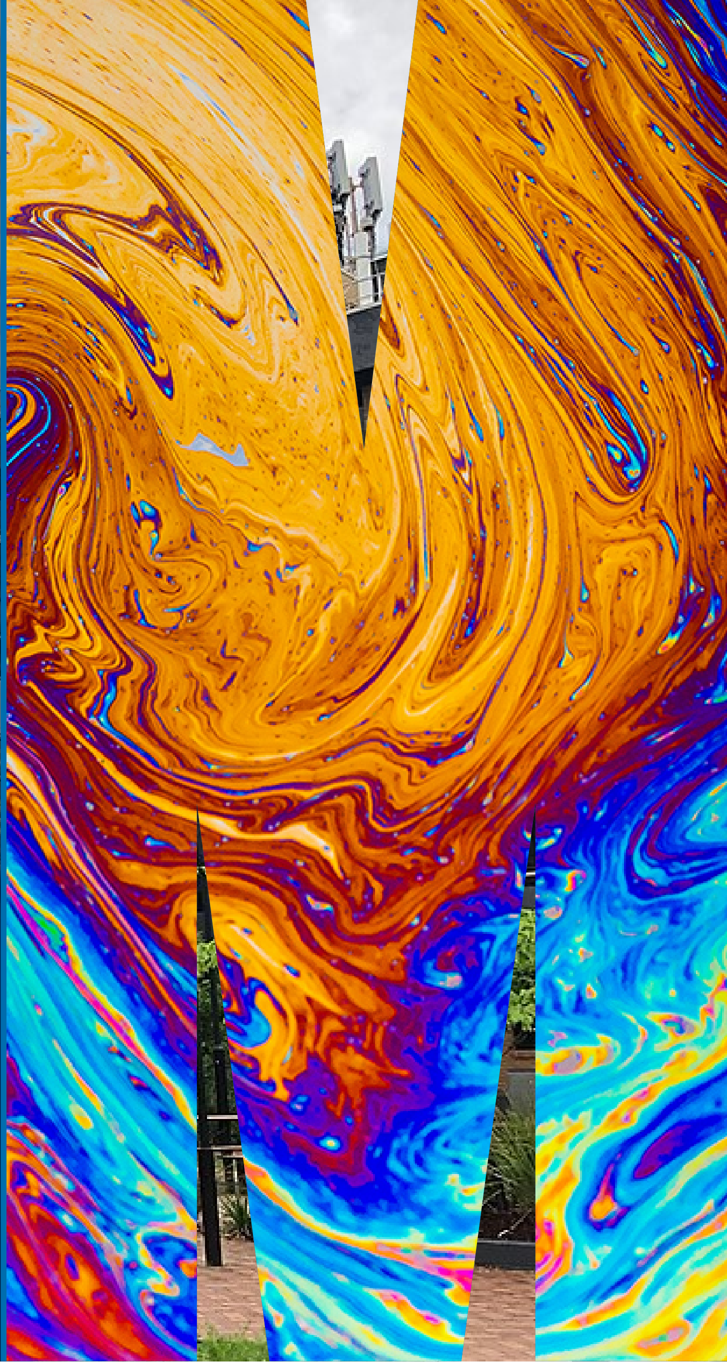
Missing Data

Lecturer: *Nicholas Tierney and Stuart Lee*

Department of Econometrics and Business Statistics

ETC5510.Clayton-x@monash.edu

April 2020



Recap

- Joins
- advanced data vis

Example

San Francisco weather data

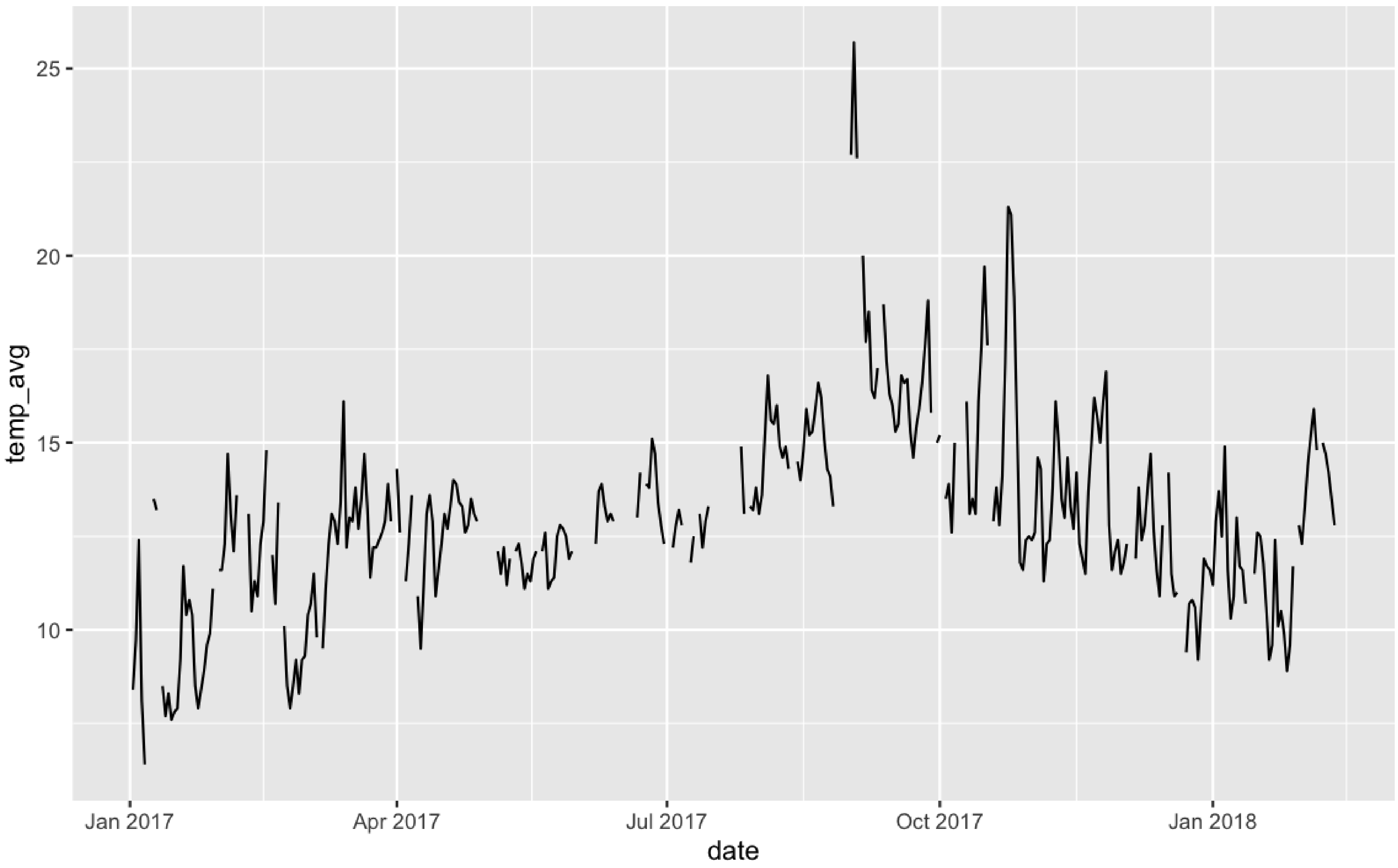
|| Date | Wind | Temp ||

Using the R package: GSODR

(Global Surface Summary of the Day)

Written by Adam Sparks

github.com/ropensci/GSODR



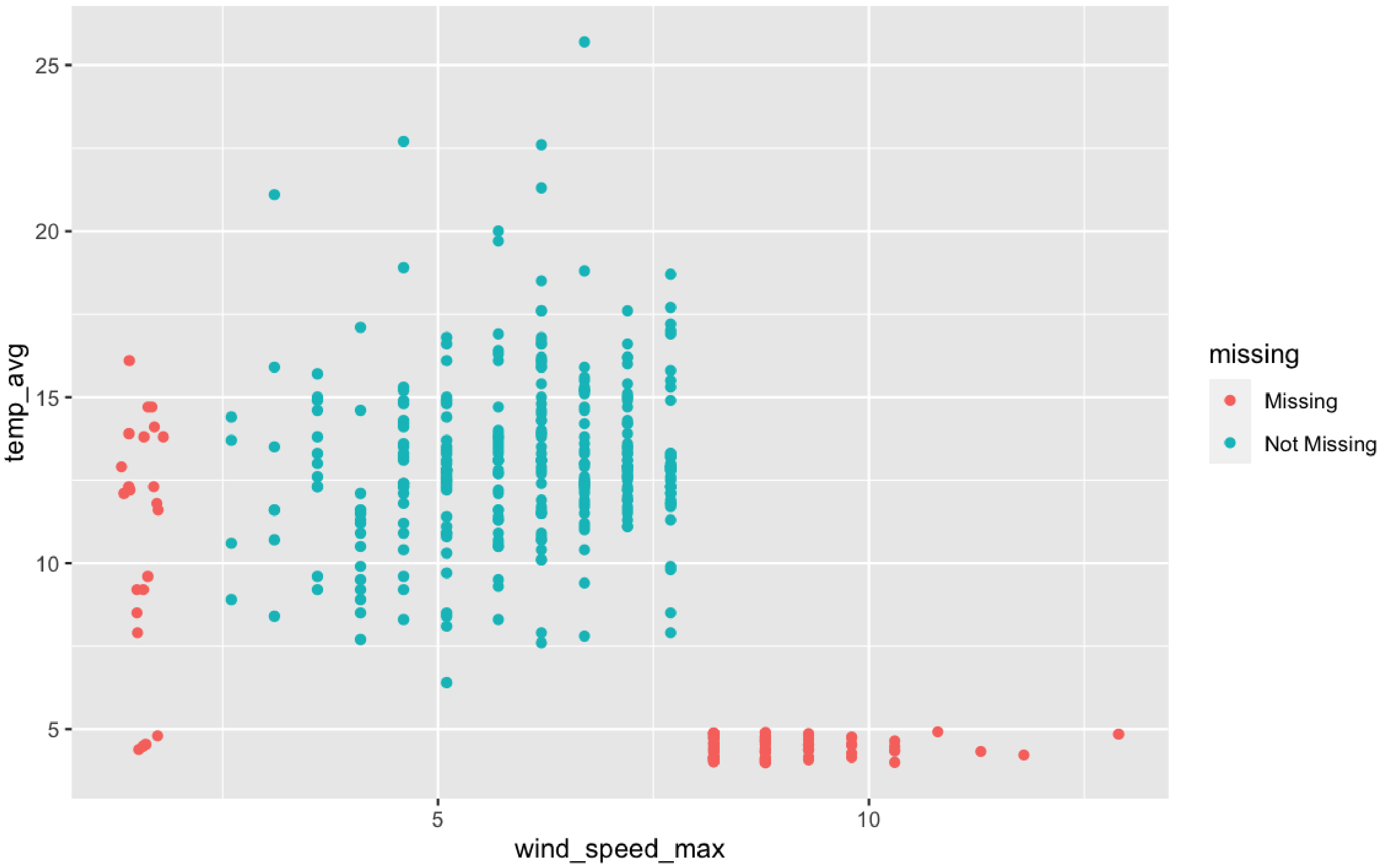
25
20
15
10
temp_avg
Jan 2017
Apr 2017
Jul 2017
Oct 2017
Jan 2018
date

Your Turn: These gaps are missing values! What are some reasons this might be a problem?

Some thoughts

- What is missing?
- Why are they missing?
- How can we summarise and explore this?

One way to show missing data



Wait, What?

What people think dealing with missing data looks like

What dealing with missing data actually looks like

What I want dealing with missing data to be like


@草根足球记录员

Learn more

visdat.njtierney.com

naniar.njtierney.com

Overview

1. What even are missing values
2. How to start looking at missing data
3. How to start exploring missing data
4. How to impute (fill in) Missing values

What are missing values?

> *Missing values are values that should have been recorded but were not.*

NA = **N**ot **A**vailable.

How do I check if I have missing values?

```
x <- c(1, NA, 3, NA, NA, 5)

library(naniar)
any_na(x)

[1] TRUE

are_na(x)

[1] FALSE  TRUE FALSE  TRUE  TRUE FALSE

n_miss(x)

[1] 3

prop_miss(x)

[1] 0.5
```

Working with missing data

NA + [anything] = NA

```
heights

Sophie    Dan   Fred
   165    177     NA

sum(heights)

[1] NA
```

Working with missing data

`na.rm = TRUE` will removes missings

```
sum(heights, na.rm = TRUE)

[1] 342
```

Use this power `responsibly`!

Dangers of removing missing values

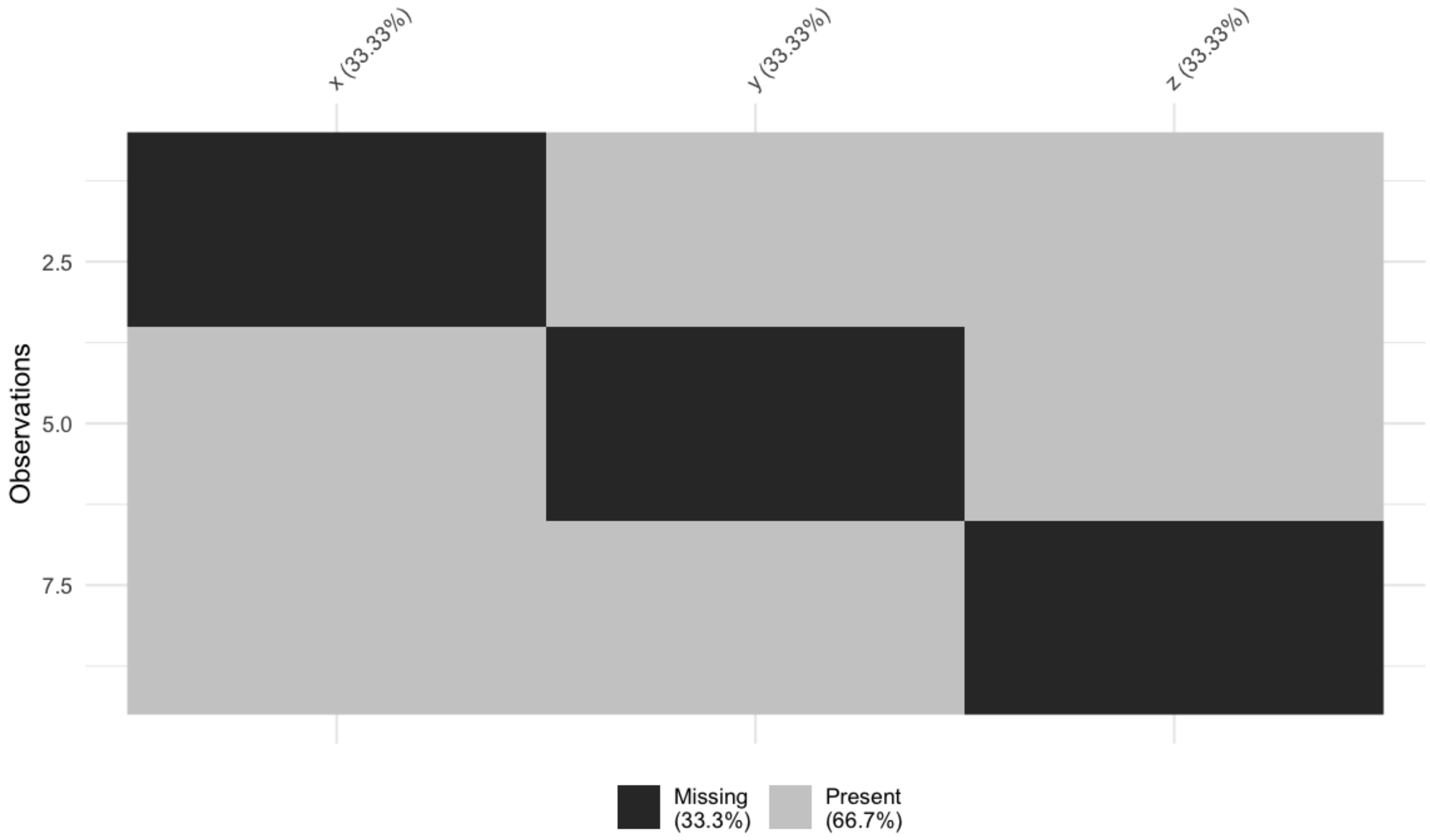
You can remove most of or all of your data:

x	y	z
NA	1	1
NA	2	2
NA	3	3
4	NA	4
5	NA	5
6	NA	6
7	7	NA
8	8	NA
9	9	NA

Dangers of removing missing values

You can remove most of or all of your data:

```
vis_miss(dat_df)
```



Dangers of removing missing values

You can remove most of or all of your data:

```
na.omit(dat_df)
## [1] x y z
## <0 rows> (or 0-length row.names)
```

wat?

`na.omit` / `na.rm` = listwise delection

x	y	z
~~NA~~	~~1~~	~~1~~
~~NA~~	~~2~~	~~2~~
~~NA~~	~~3~~	~~3~~
~~4~~	~~NA~~	~~4~~
~~5~~	~~NA~~	~~5~~
~~6~~	~~NA~~	~~6~~
~~7~~	~~7~~	~~NA~~
~~8~~	~~8~~	~~NA~~
~~9~~	~~9~~	~~NA~~

`na.omit` / `na.rm` = listwise delection

x	y	z
~~NA~~	~~1~~	~~1~~
~~NA~~	~~2~~	~~2~~
~~NA~~	~~3~~	~~3~~
4	NA	4
5	NA	5
6	NA	6
7	7	NA
8	8	NA
9	9	NA

`na.omit` / `na.rm` = listwise delection

x	y	z
~~NA~~	~~1~~	~~1~~
~~NA~~	~~2~~	~~2~~
~~NA~~	~~3~~	~~3~~
~~4~~	~~NA~~	~~4~~
~~5~~	~~NA~~	~~5~~
~~6~~	~~NA~~	~~6~~
7	7	NA
8	8	NA
9	9	NA

`na.omit` / `na.rm` = listwise delection

x	y	z
~~NA~~	~~1~~	~~1~~
~~NA~~	~~2~~	~~2~~
~~NA~~	~~3~~	~~3~~
~~4~~	~~NA~~	~~4~~
~~5~~	~~NA~~	~~5~~
~~6~~	~~NA~~	~~6~~
~~7~~	~~7~~	~~NA~~
~~8~~	~~8~~	~~NA~~
~~9~~	~~9~~	~~NA~~

Takehome:

- na.rm or na.omit can remove entire rows containing missings
- This is bad because you can lose data - sometimes all your data! This might not be what you anticipate!
- It can also mean that you are removing / censoring observations.

Dangers of removing missing values

You can introduce bias - what happens when you remove the NAs?

temp	location
27	inside
26	inside
NA	outside
29	inside
NA	outside
20	outside
21	outside
24	inside

Your turn:

- Open rstudio.
- go to `exercise-5a-intro-missing.Rmd`
- type the following:

```r
# install.packages("usethis")
library(usethis)
use_course("https://mida.numbat.space/exercises/5a/mida-exercise-5a.zip")
```

Introduction to missingness summaries

Basic summaries of missingness:

- `n_miss`
- `n_complete`

Dataframe summaries of missingness:

- `miss_var_summary`
- `miss_case_summary`

These functions work with `group_by`

Missing data summaries: Variables

```
miss_var_summary(dat_sf_clean)
## # A tibble: 6 x 3
##   variable       n_miss pct_miss
##   <chr>           <int>    <dbl>
## 1 temp_min           70    17.3
## 2 temp_max           70    17.3
## 3 temp_avg           70    17.3
## 4 wind_speed_max     23     5.68
## 5 date                0     0
## 6 month               0     0
```

Missing data summaries: Cases

```
miss_case_summary(dat_sf_clean)
## # A tibble: 405 x 3
##     case n_miss pct_miss
##    <int>  <int>    <dbl>
##  1    89      4     66.7
##  2   182      4     66.7
##  3   188      4     66.7
##  4   271      4     66.7
##  5     6      3     50
##  6     7      3     50
##  7    10      3     50
##  8    29      3     50
##  9    37      3     50
## 10    39      3     50
## # … with 395 more rows
```

Missing data tabulations: variables

```r
miss_var_table(dat_sf_clean)
## # A tibble: 3 x 3
##   n_miss_in_var n_vars pct_vars
##           <int>  <int>    <dbl>
## 1             0      2     33.3
## 2            23      1     16.7
## 3            70      3     50
```

Missing data tabulations: cases

```
miss_case_table(dat_sf_clean)
## # A tibble: 4 x 3
##   n_miss_in_case n_cases pct_cases
##            <int>   <int>     <dbl>
## 1              0     316    78.0
## 2              1      19     4.69
## 3              3      66    16.3
## 4              4       4     0.988
```

Using summaries with `group_by`

```
dat_sf_clean %>%
  group_by(month) %>%
  miss_var_summary()
## # A tibble: 60 x 4
## # Groups:   month [12]
##    month variable       n_miss pct_miss
##    <dbl> <chr>           <int>    <dbl>
##  1     1 temp_min            7    11.5
##  2     1 temp_max            7    11.5
##  3     1 temp_avg            7    11.5
##  4     1 wind_speed_max      4     6.56
##  5     1 date                0     0
##  6     2 temp_min            5    12.8
##  7     2 temp_max            5    12.8
##  8     2 temp_avg            5    12.8
##  9     2 wind_speed_max      4    10.3
## 10     2 date                0     0
## # … with 50 more rows
```

Your Turn

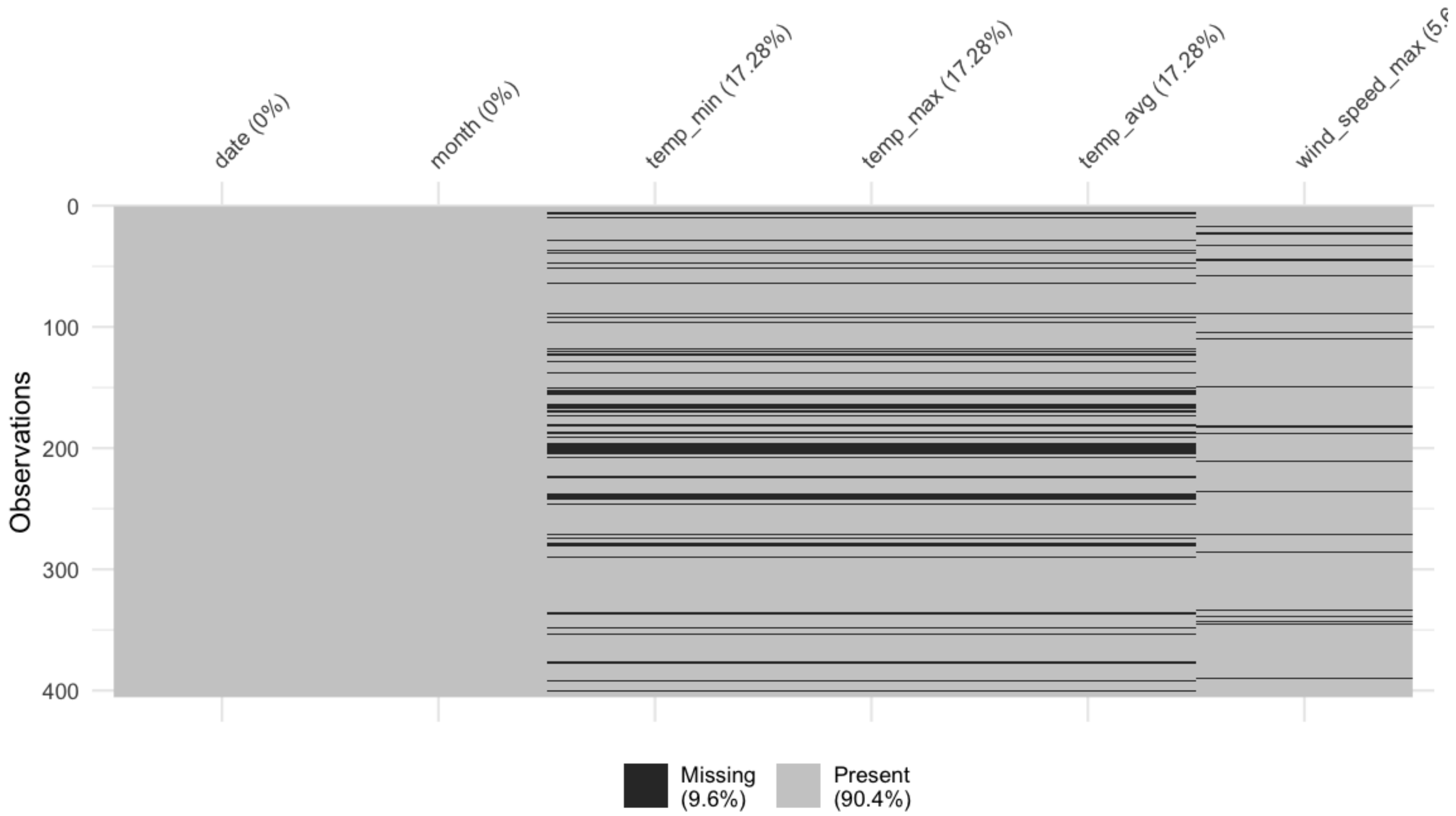
Open exercise-5a-summarise-missings.Rmd

Introduction to missing data visualisations in naniar

- Visualisation can quickly capture an idea or thought.
- `naniar` provides a friendly family of missing data visualization functions.
- Each visualization corresponds to a data summary.
- Visualisations help you operate closer to the speed of thought.

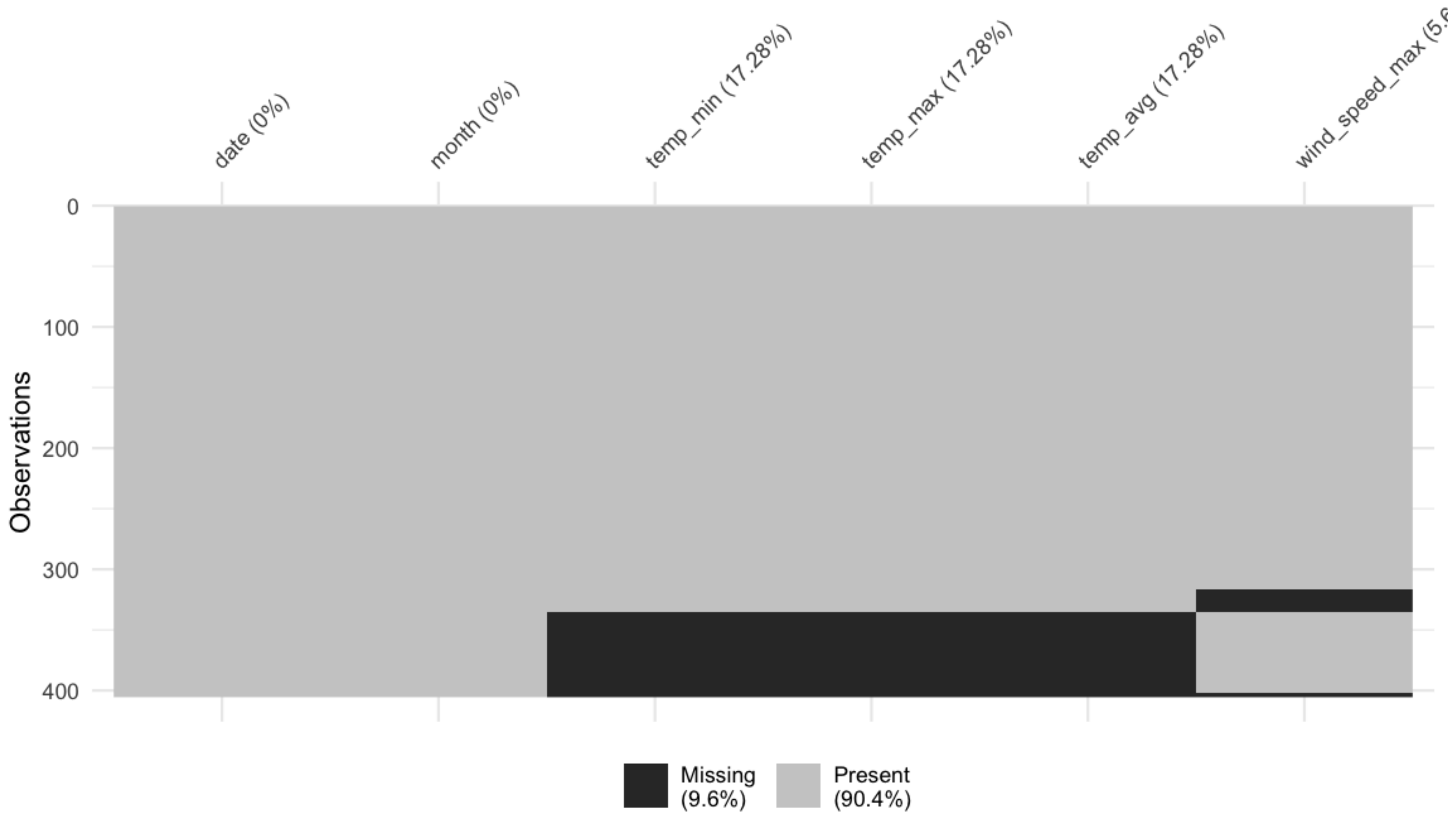
Get a bird's eye view of the missing data

```
vis_miss(dat_sf_clean)
```



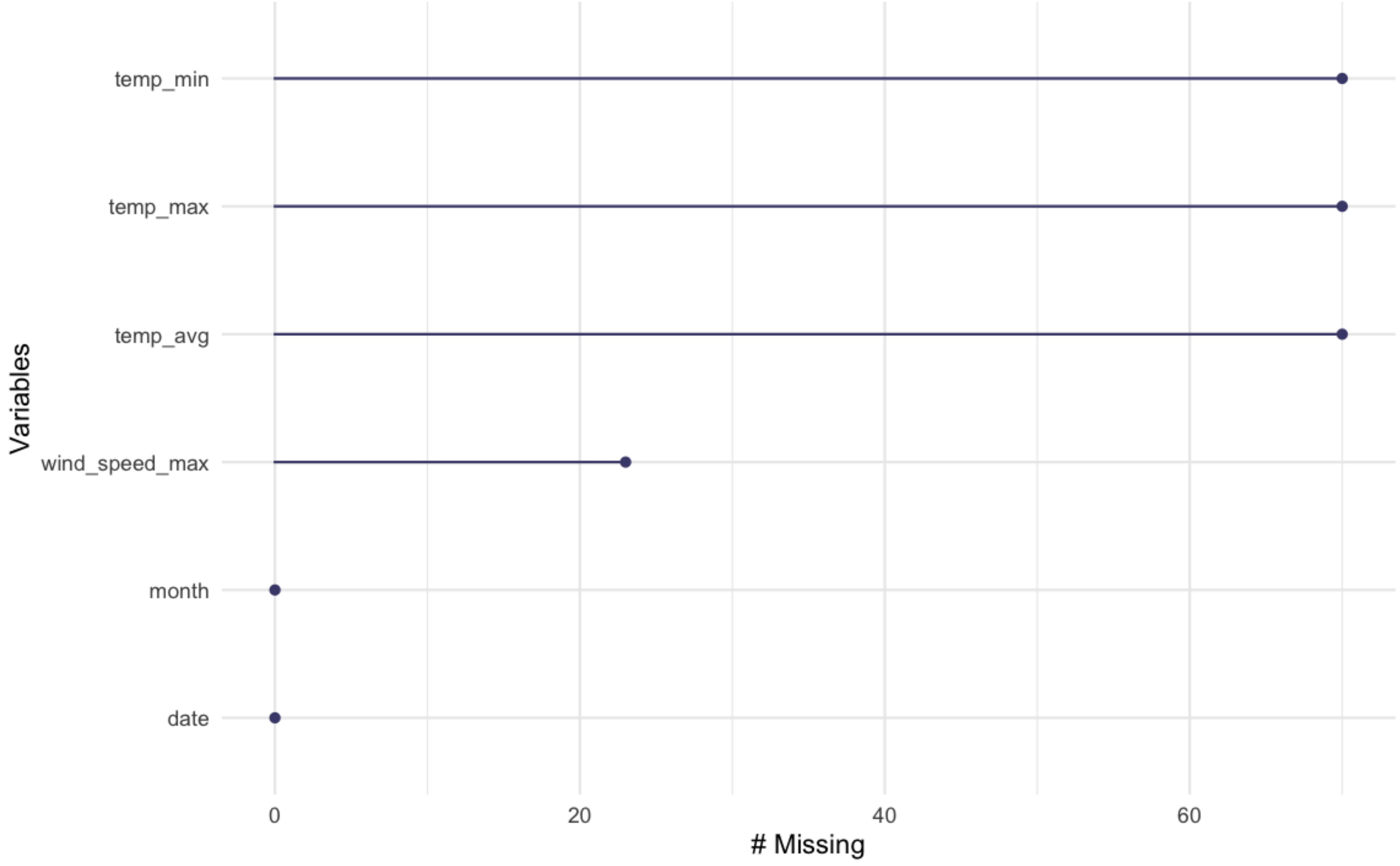
Get a bird's eye view of the missing data

```
vis_miss(dat_sf_clean, cluster = TRUE)
```



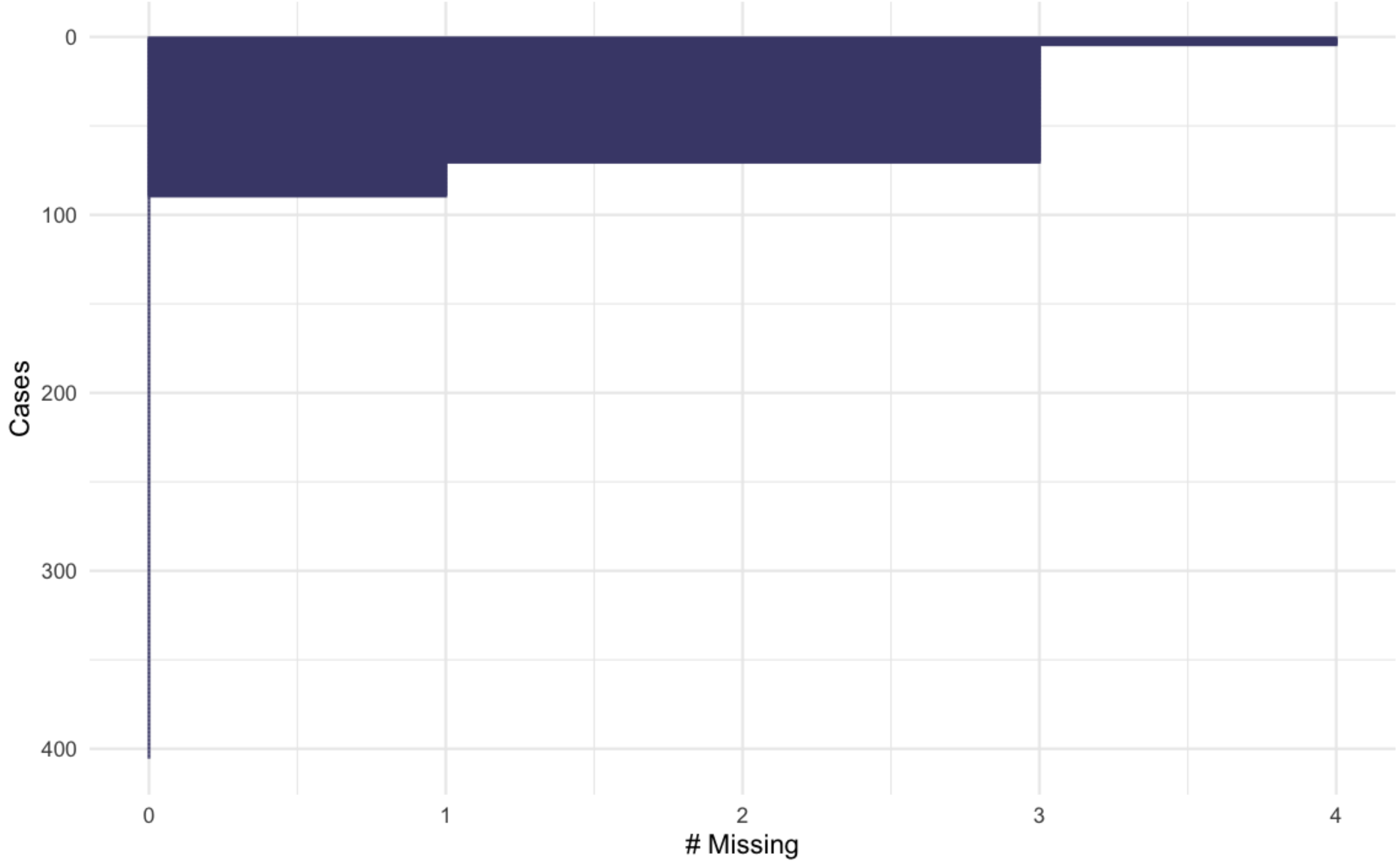
Look at missings in cases

```
gg_miss_var(dat_sf_clean)
```



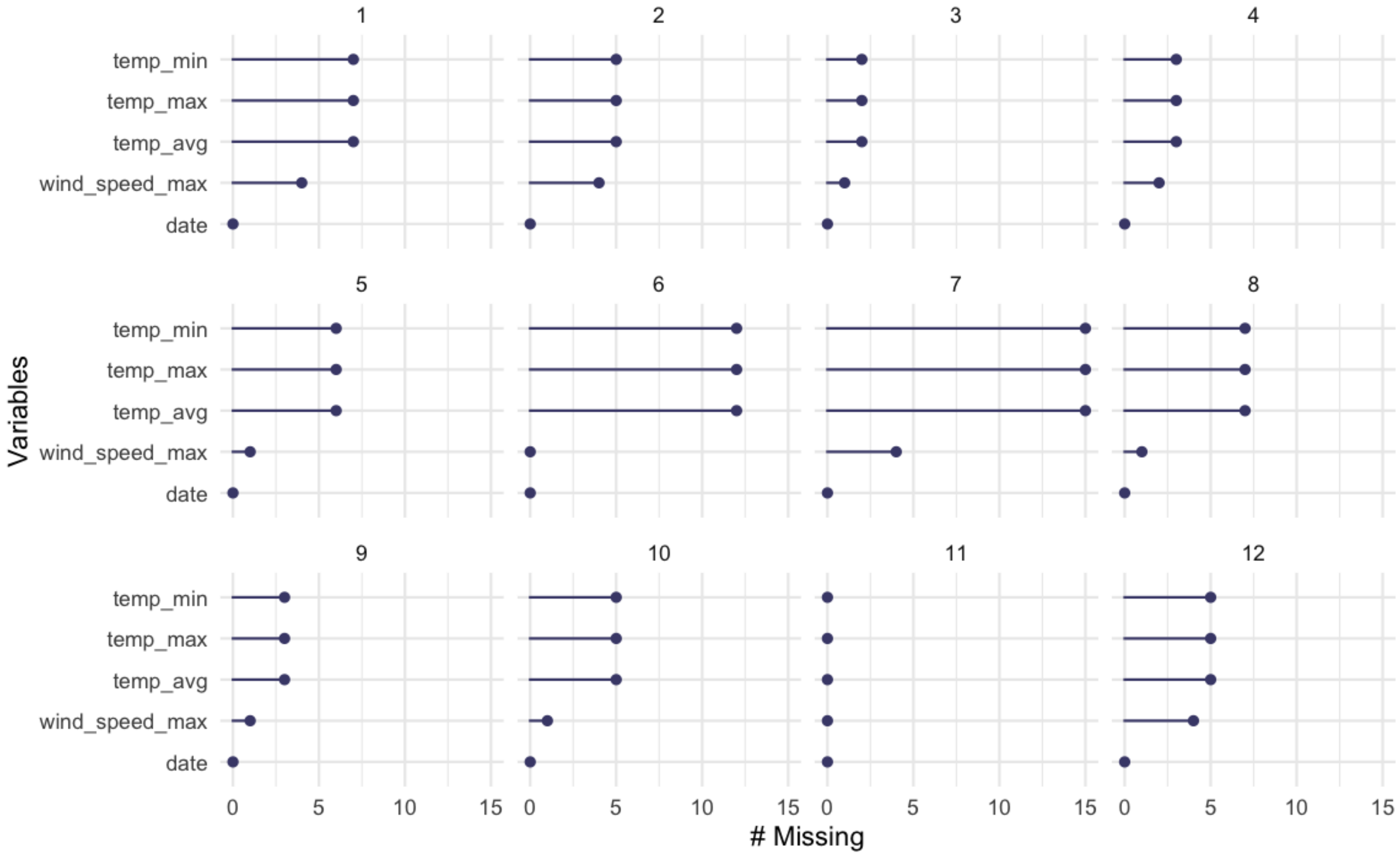
Look at missings in cases

```
gg_miss_case(dat_sf_clean)
```



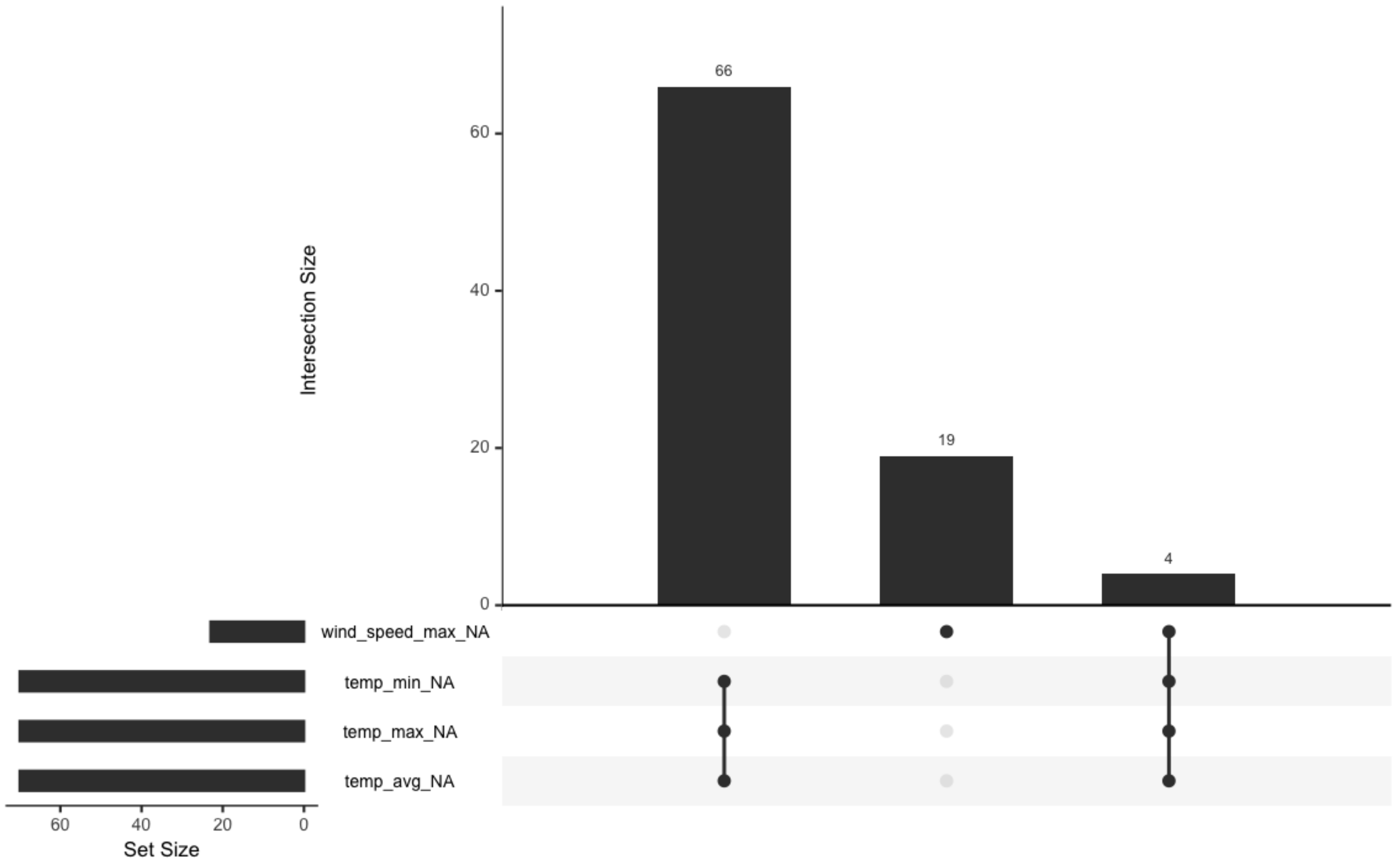
Look at missings in variables

```
gg_miss_var(dat_sf_clean, facet = month)
```



Visualizing missingness patterns

```
gg_miss_upset(dat_sf_clean)
```



Your turn

- complete exercise-5a-visualise-missings.Rmd

miss_*	gg_miss_*
miss_var_*	gg_miss_var
miss_case_*	gg_miss_case

Representing *Missing* values in a *Tidy* Way

Tidy Data

Variables in columns

Observations in Rows

One value per cell

A	B
2018	NA
NA	“Sam”

Data Shadow

Variable ends in NA

Values are missing (NA) or not (!NA)

A_NA	B_NA
!NA	NA
NA	!NA

Tidy Missing Data

`bind_shadow(data)`

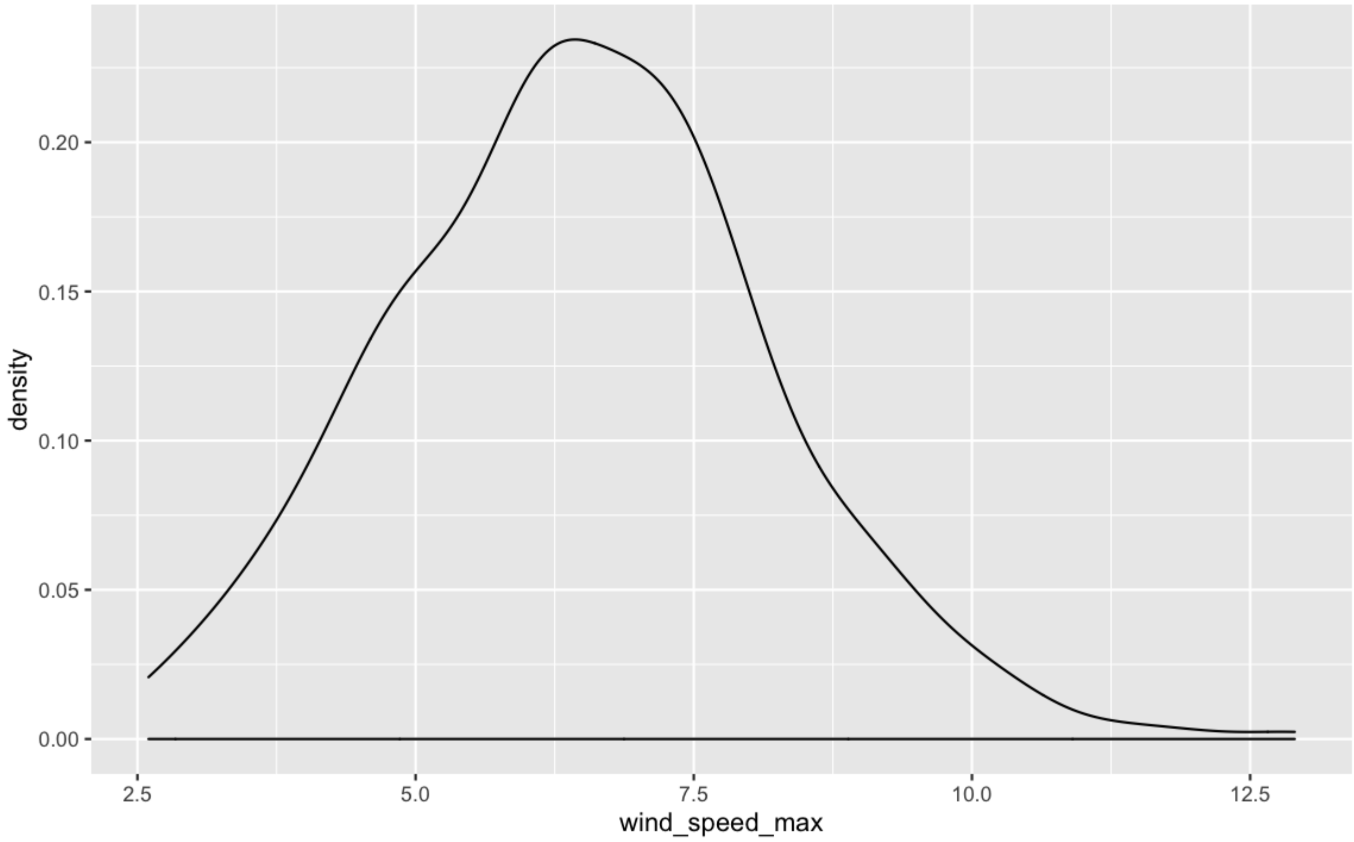
A	B	A_NA	B_NA
2018	NA	!NA	NA
NA	“Sam”	NA	!NA

bind_shadow()

```
bind_shadow(dat_sf_clean) %>% glimpse()
## Rows: 405
## Columns: 12
## $ date              <date> 2017-01-02, 2017-01-03, 2017-01-04, 2017-01-05, 20…
## $ month             <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ temp_min          <dbl> 5.9, 8.4, 10.4, 5.8, 4.4, NA, NA, 11.9, 11.9, NA, 6…
## $ temp_max          <dbl> 10.5, 11.1, 14.5, 9.4, 9.1, NA, NA, 16.0, 14.6, NA,…
## $ temp_avg          <dbl> 8.4, 9.7, 12.4, 8.1, 6.4, NA, NA, 13.5, 13.2, NA, 8…
## $ wind_speed_max    <dbl> 5.1, 5.1, 6.7, 5.1, 5.1, 8.8, 8.2, 7.2, 7.7, 8.2, 5…
## $ date_NA           <fct> !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !…
## $ month_NA          <fct> !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !…
## $ temp_min_NA       <fct> !NA, !NA, !NA, !NA, !NA, NA, NA, !NA, !NA, NA, !NA,…
## $ temp_max_NA       <fct> !NA, !NA, !NA, !NA, !NA, NA, NA, !NA, !NA, NA, !NA,…
## $ temp_avg_NA       <fct> !NA, !NA, !NA, !NA, !NA, NA, NA, !NA, !NA, NA, !NA,…
## $ wind_speed_max_NA <fct> !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !NA, !…
```

Shadows In Practice: Explore one variable

```
dat_sf_clean %>%
  ggplot(aes(x = wind_speed_max)) +
  geom_density()
```

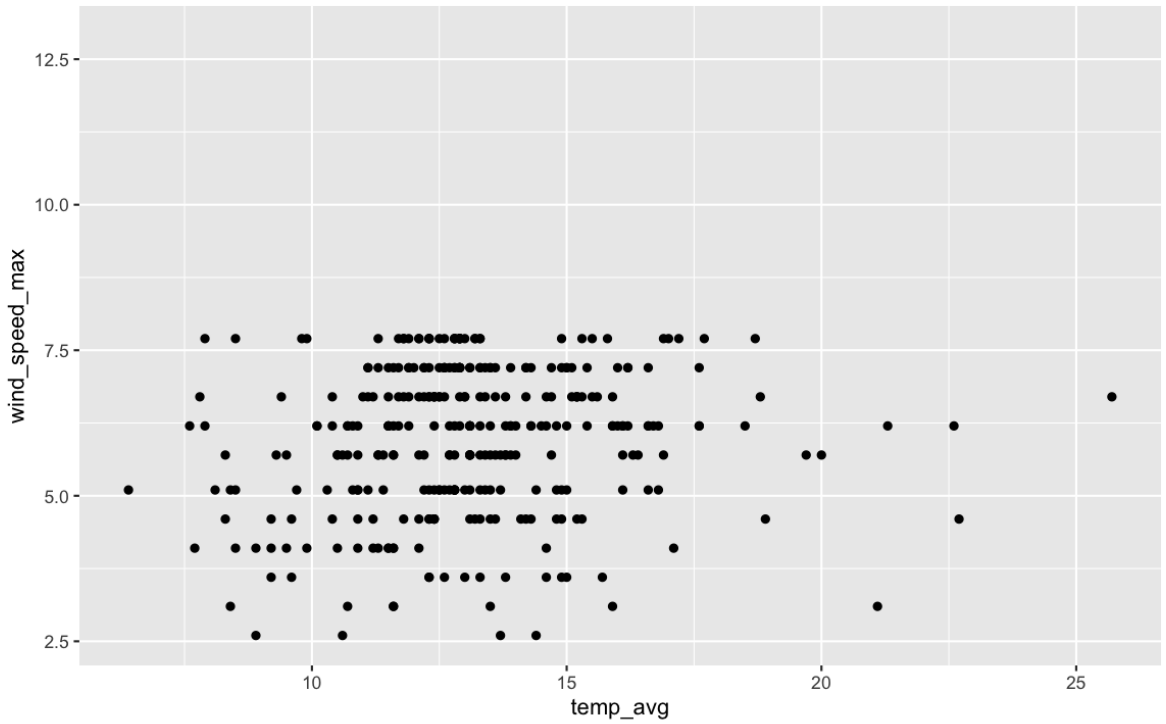


Shadows In Practice: Explore one variable

```
dat_sf_clean %>%
bind_shadow() %>%
  ggplot(aes(x = wind_speed_max,
             colour = temp_avg_NA)) +
  geom_density()
```

In Practice: Explore two variables

```
ggplot(dat_sf_clean,
       aes(x = temp_avg,
           y = wind_speed_max))
  geom_point()
```



Impute shadow values into our realm

```
## # A tibble: 7 x 2
##   temp_avg temp_avg_NA
##      <dbl> <fct>
## 1       8.4 !NA
## 2       9.7 !NA
## 3      12.4 !NA
## 4       8.1 !NA
## 5       6.4 !NA
## 6      NA   NA
## 7      NA   NA
```

```
## # A tibble: 7 x 2
##   temp_avg temp_avg_NA
##      <dbl> <fct>
## 1       8.4   !NA
## 2       9.7   !NA
## 3      12.4   !NA
## 4       8.1   !NA
## 5       6.4   !NA
## 6       5.66 NA
## 7       5.69 NA
```

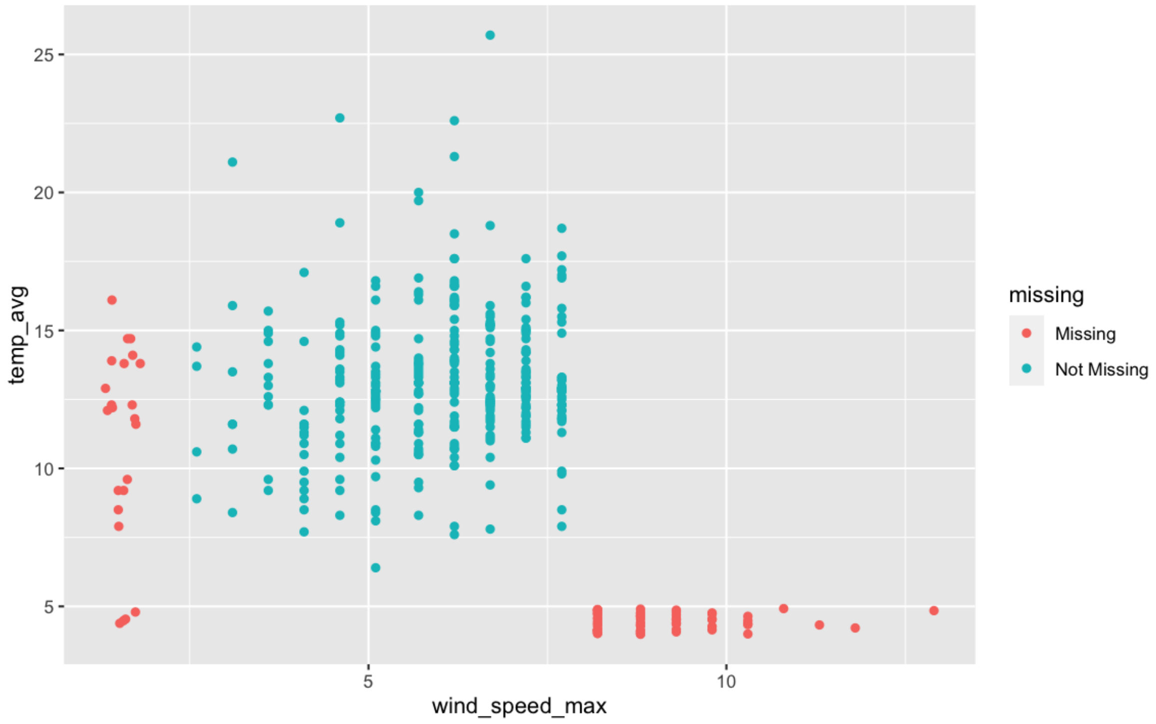
impute_below()

Impute missing values from the shadows into our realm

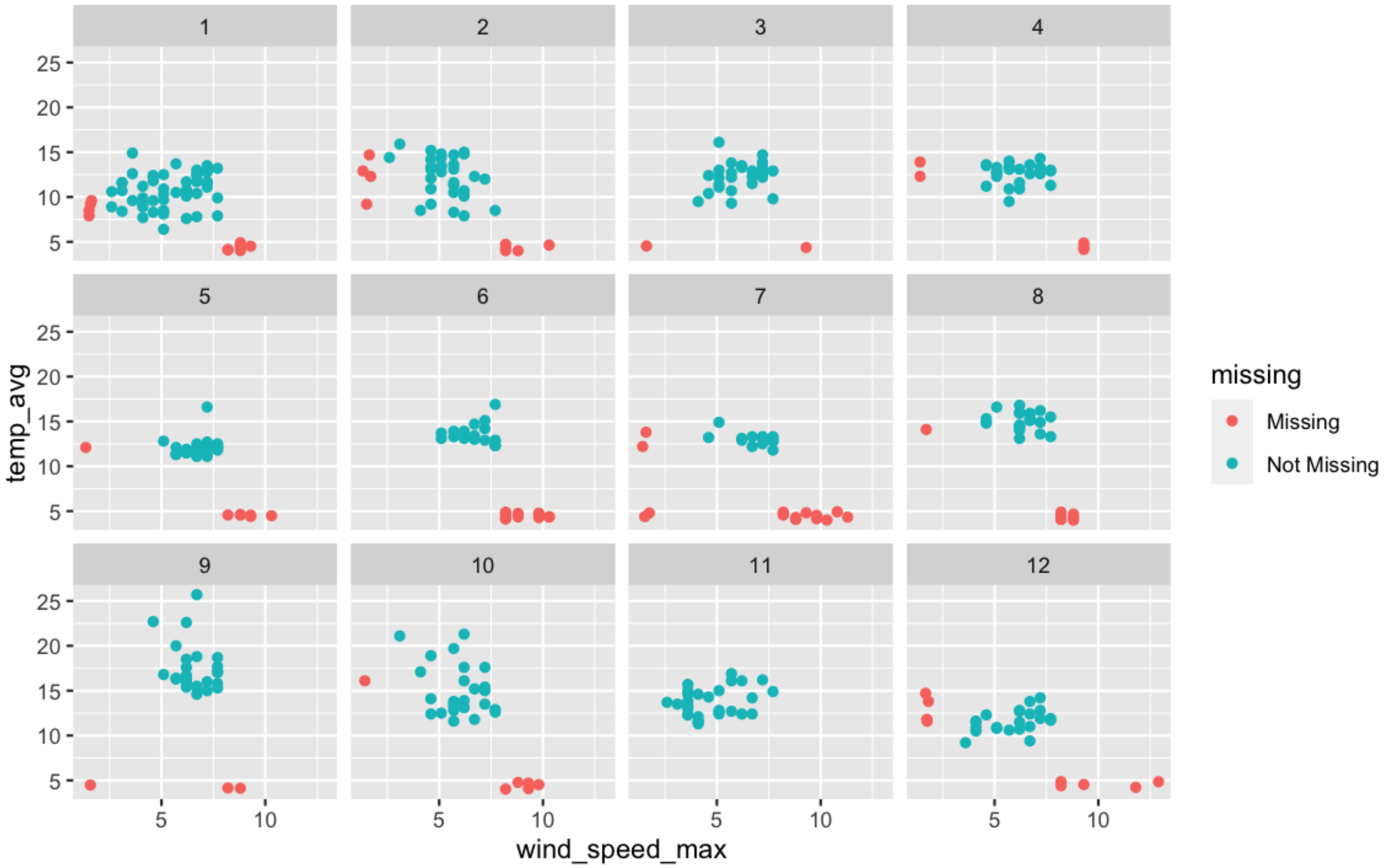
```
dat_sf_clean %>%
  slice(5:10) %>%
  mutate(temp_avg_shift = impute_below(temp_avg)) %>%
  select(temp_avg, temp_avg_shift)
## # A tibble: 6 x 2
##   temp_avg temp_avg_shift
##      <dbl>          <dbl>
## 1      6.4           6.4
## 2     NA             5.73
## 3     NA             5.74
## 4     13.5          13.5
## 5     13.2          13.2
## 6     NA             5.53
```

geom_miss_point()

```
ggplot(dat_sf_clean,
       aes(x = wind_speed_max,
           y = temp_avg)) +
  geom_miss_point()
```



Facets!

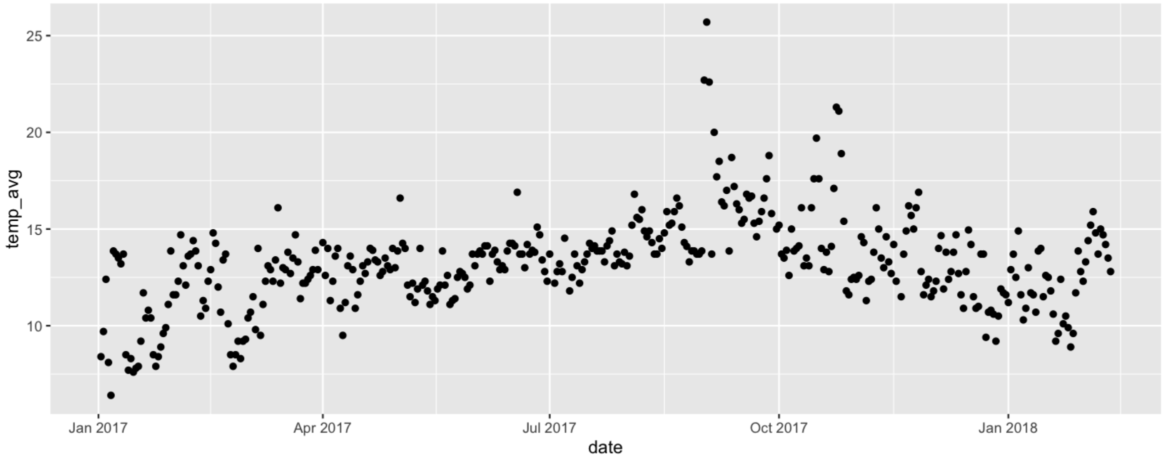

1
2
3
4
5
6
7
8
9
10
11
12
25
20
15
10
5
temp_avg
wind_speed_max
missing
Missing
Not Missing

Exploring imputed values

Imputation is the process of filling in missing values with some other estimate

What about this imputation thing?

```
dat_sf_clean %>%
  as.data.frame() %>%
  simputation::impute_lm(temp_avg ~ wind_speed_max) %>%
  ggplot(aes(x = date,
             y = temp_avg)) +
  geom_point()
```



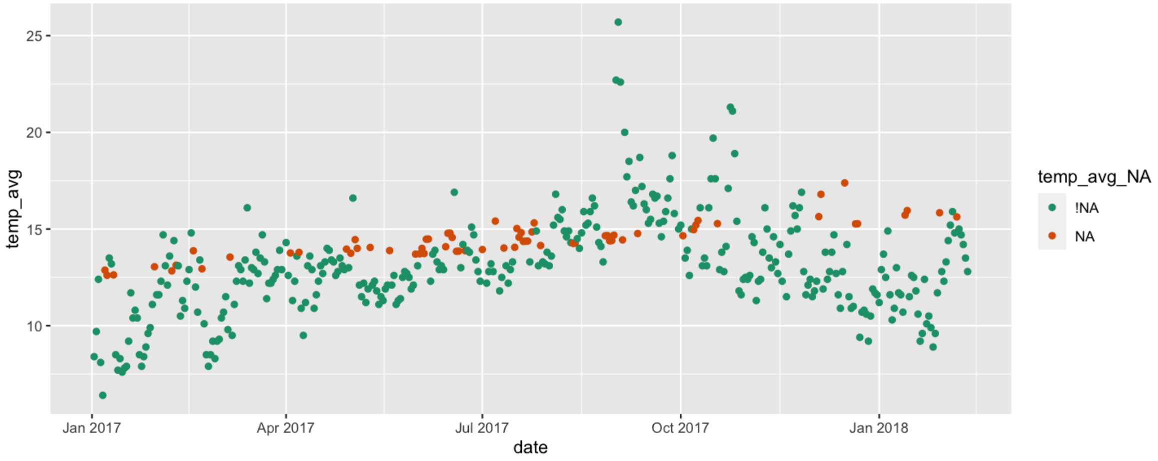
What about this imputation thing?

They are **invisible!**

Where are the imputed values?

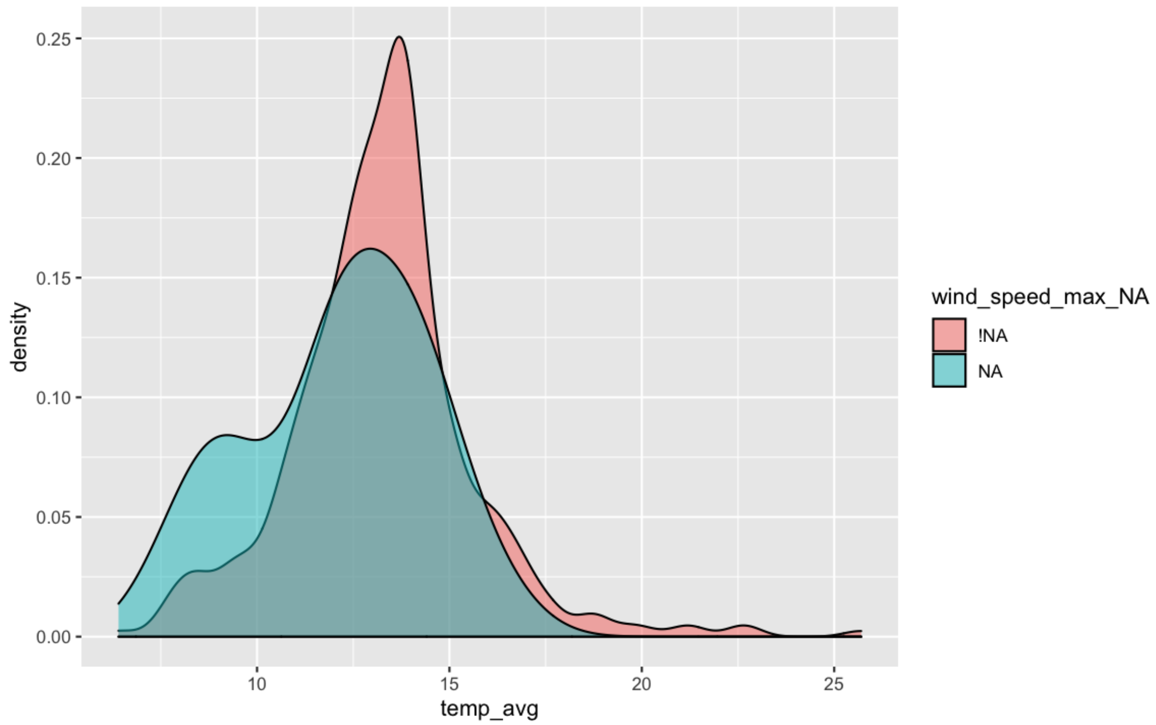
Tidy Missing Data reveals the imputations!

```
bind_shadow(dat_sf_clean) %>%
  as.data.frame() %>%
  simputation::impute_lm(temp_avg ~ wind_speed_max + date) %>%
  as_tibble() %>%
  ggplot(aes(x = date,
             y = temp_avg,
             colour  = temp_avg_NA)) +
  geom_point() +
  scale_colour_brewer(palette = "Dark2")
```



Shadows make things clearer!

```
bind_shadow(dat_sf_clean) %>%
  as.data.frame() %>%
  simputation::impute_lm(temp_avg ~ wind_speed_max) %>%
  ggplot(aes(x = temp_avg,
             fill = wind_speed_max_NA)) +
  geom_density(alpha = 0.5)
```

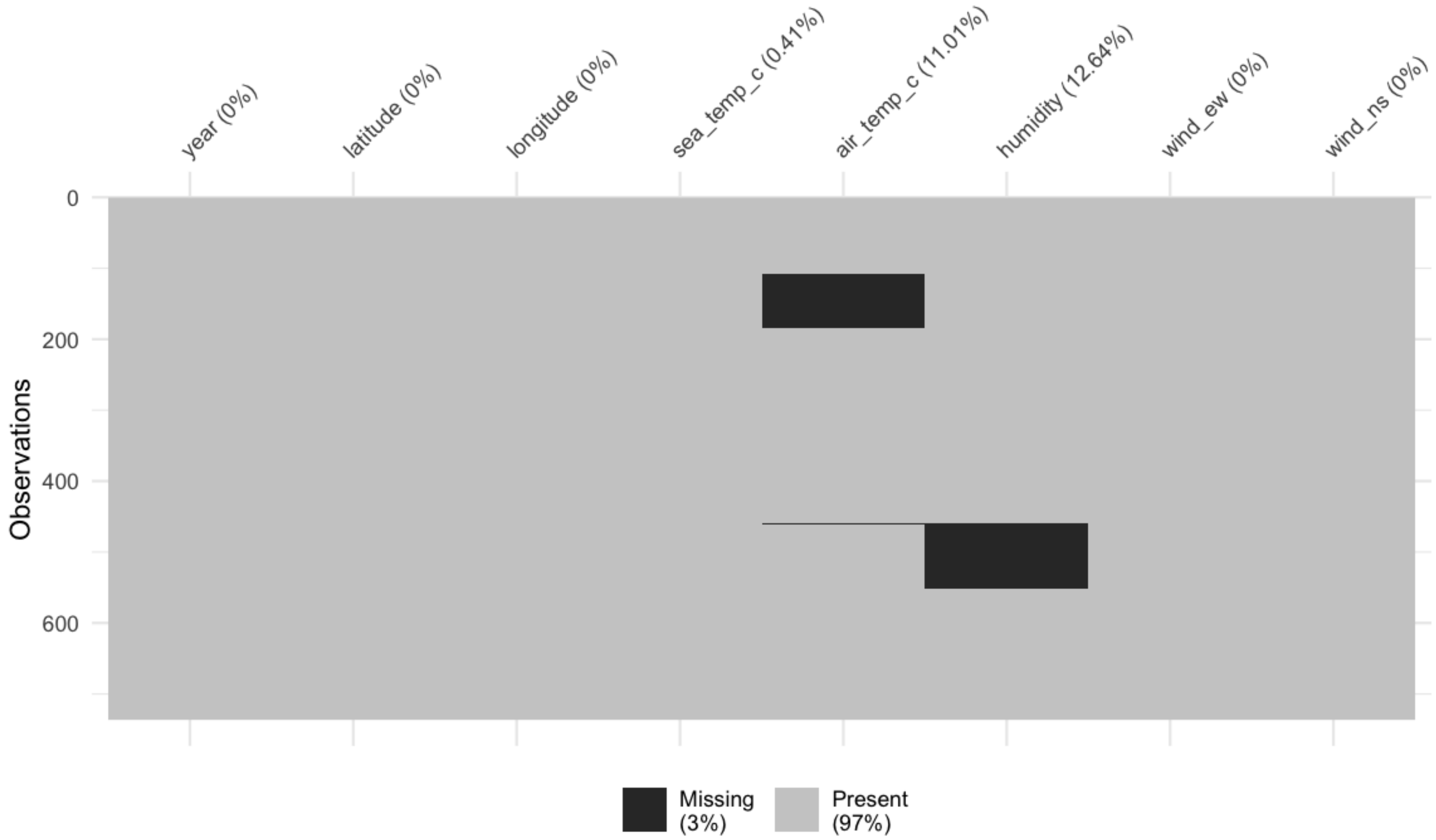


Example data: oceanbuoys

```
oceanbuoys
## # A tibble: 736 x 8
##     year latitude longitude sea_temp_c air_temp_c humidity wind_ew wind_ns
##    <dbl>    <dbl>     <dbl>      <dbl>      <dbl>    <dbl>   <dbl>   <dbl>
##  1  1997        0      -110       27.6       27.1     79.6   -6.40    5.40
##  2  1997        0      -110       27.5       27.0     75.8   -5.30    5.30
##  3  1997        0      -110       27.6       27       76.5   -5.10    4.5
##  4  1997        0      -110       27.6       26.9     76.2   -4.90    2.5
##  5  1997        0      -110       27.6       26.8     76.4   -3.5     4.10
##  6  1997        0      -110       27.8       26.9     76.7   -4.40    1.60
##  7  1997        0      -110       28.0       27.0     76.5   -2       3.5
##  8  1997        0      -110       28.0       27.1     78.3   -3.70    4.5
##  9  1997        0      -110       28.0       27.2     78.6   -4.20    5
## 10  1997        0      -110       28.0       27.2     76.9   -3.60    3.5
## # … with 726 more rows
```

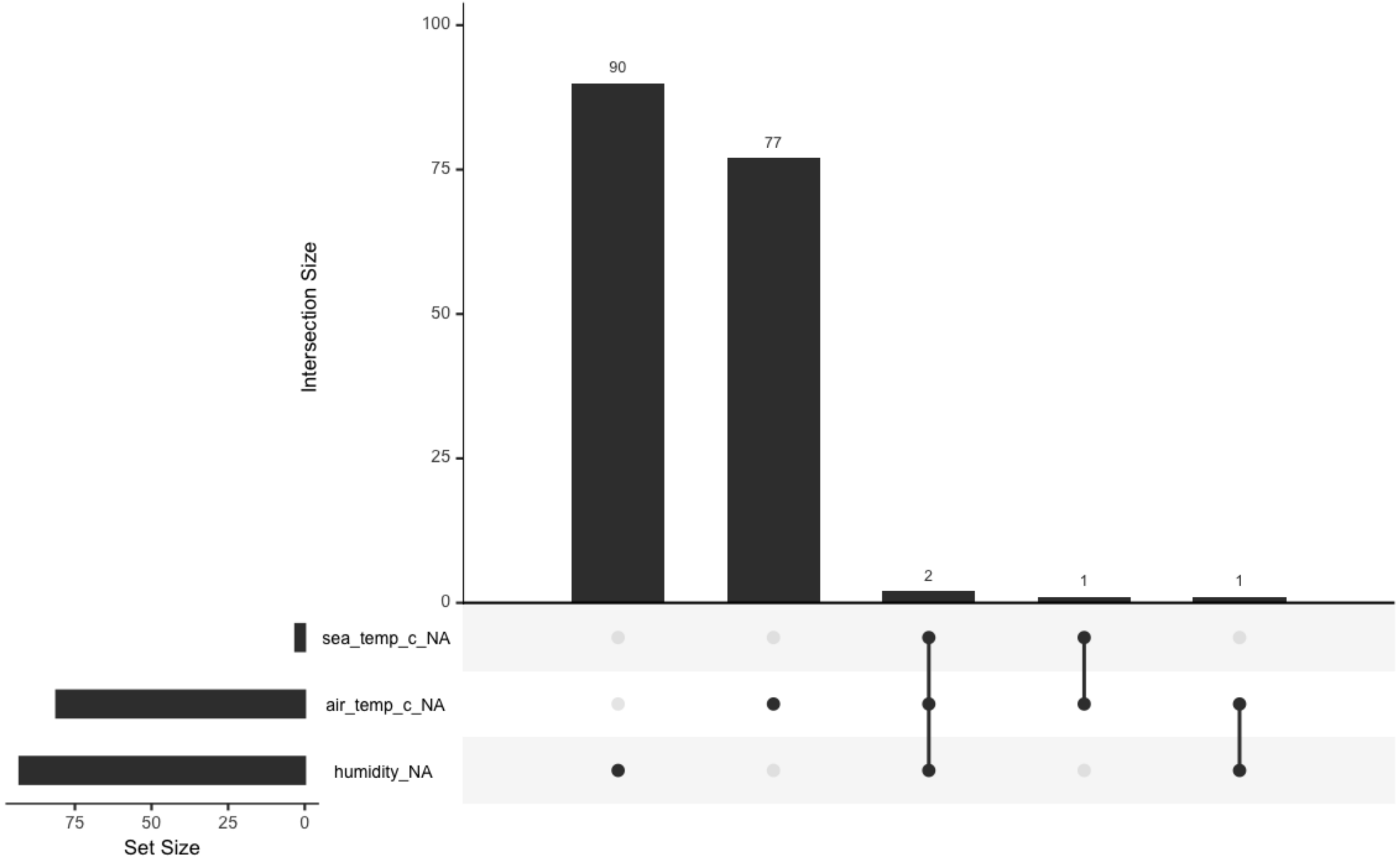
Start looking at missing values

```
vis_miss(oceanbuoys)
```



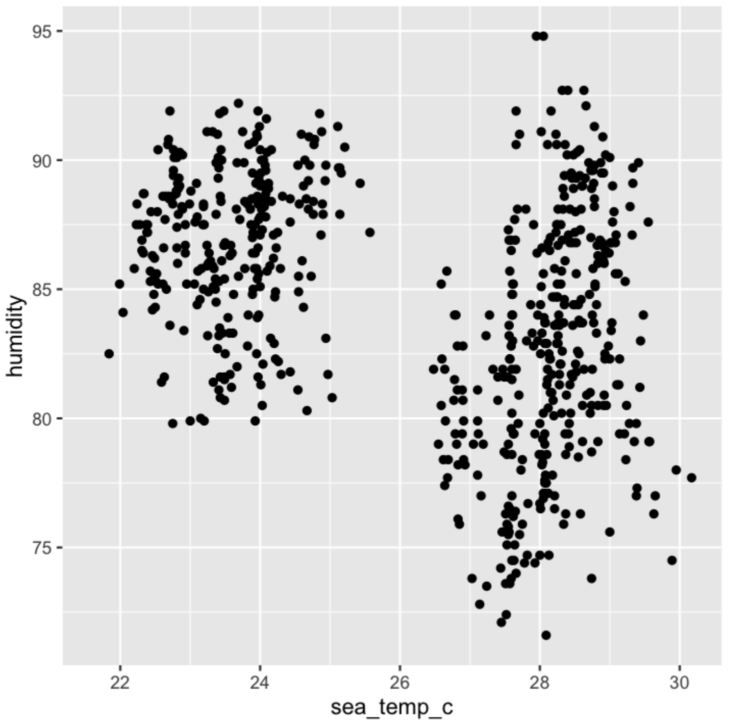
Missing value patterns

```
gg_miss_upset(oceanbuoys)
```



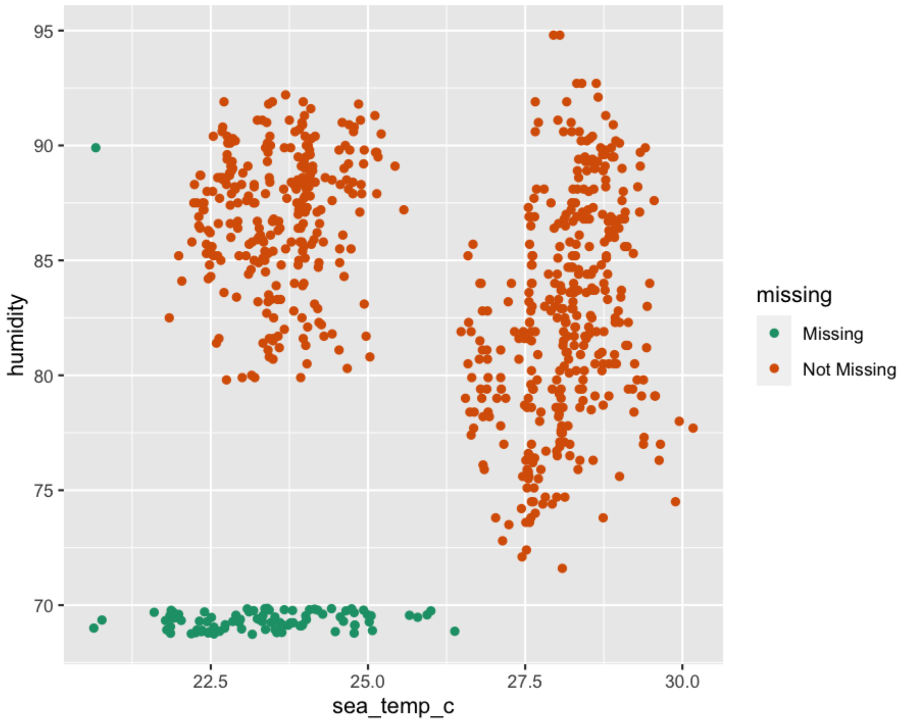
Missings Tend to get ignored by most software

```
ggplot(oceanbuoys,
  aes(x = sea_temp_c,
      y = humidity)) +
  geom_point() +
  theme(aspect.ratio = 1)
```



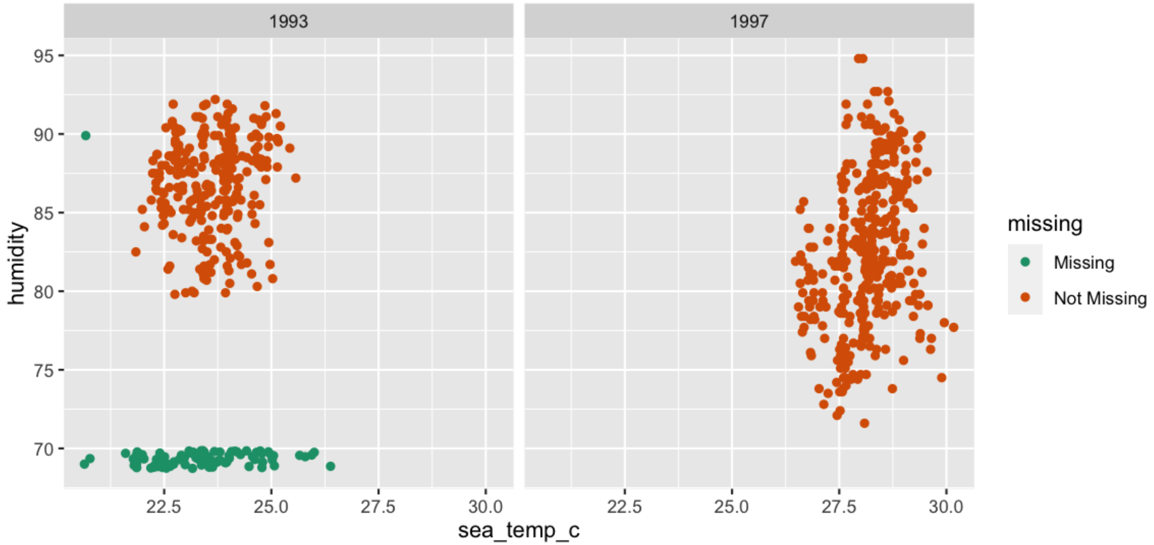
Add missings to plot with geom_miss_point()

```
ggplot(oceanbuoys,
       aes(x = sea_temp_c,
           y = humidity)) +
  scale_colour_brewer(palette="[
  geom_miss_point() + theme(aspe
```



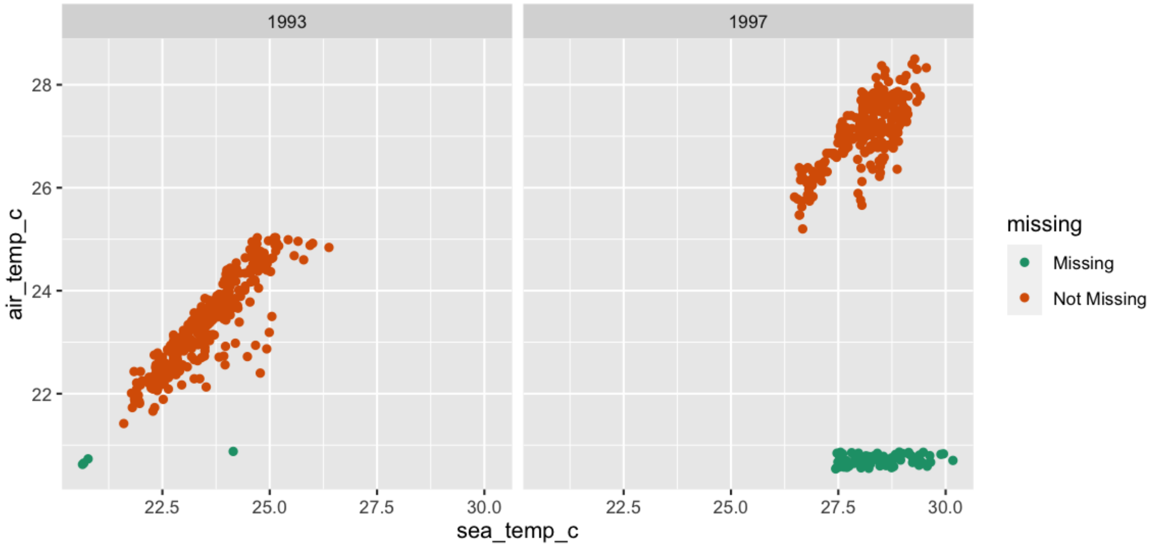
Facet By year

```
ggplot(oceanbuoys,
       aes(x = sea_temp_c, y = humidity)) +
  geom_miss_point() +
  scale_colour_brewer(palette = "Dark2") +
  facet_wrap(~year) +
  theme(aspect.ratio=1)
```



Understanding missing dependencies

```
ggplot(oceanbuoys,
       aes(x = sea_temp_c,
           y = air_temp_c)) +
  geom_miss_point() +
  scale_colour_brewer(palette="[
  facet_wrap(~year) +
  theme(aspect.ratio=1)
```



Strategies for working with missing values

- Small fraction of cases have several missings (around 5%) - explore data, and possibly drop the cases
- A variable or two, out of many, have a lot of missings, drop the variables

Strategies for working with missing values

- If missings are small in number, but located in many cases and variables, you need to impute these values, to do most analyses
- Designing the imputation should take into account dependencies that you have seen between missingness and existing variables.
- For the ocean buoys data this means imputation needs to be done separately by year

Common ways to impute values

- (Usually bad) Simple parametric: use the mean or median of the complete cases for each variable
- (Better) More complex: use models to predict missing values
- (Best) Multiple imputation: Use a statistical distribution, e.g. normal model and simulate a value (or set of values, hot deck imputation) for the missings.

Setup for missings

```
tao_shadow <- bind_shadow(oceanbuoys)

tao_shadow
## # A tibble: 736 x 16
##     year latitude longitude sea_temp_c air_temp_c humidity wind_ew wind_ns
##    <dbl>    <dbl>     <dbl>      <dbl>      <dbl>    <dbl>   <dbl>   <dbl>
##  1  1997        0      -110       27.6       27.1     79.6   -6.40    5.40
##  2  1997        0      -110       27.5       27.0     75.8   -5.30    5.30
##  3  1997        0      -110       27.6       27       76.5   -5.10    4.5
##  4  1997        0      -110       27.6       26.9     76.2   -4.90    2.5
##  5  1997        0      -110       27.6       26.8     76.4   -3.5     4.10
##  6  1997        0      -110       27.8       26.9     76.7   -4.40    1.60
##  7  1997        0      -110       28.0       27.0     76.5   -2       3.5
##  8  1997        0      -110       28.0       27.1     78.3   -3.70    4.5
##  9  1997        0      -110       28.0       27.2     78.6   -4.20    5
## 10  1997        0      -110       28.0       27.2     76.9   -3.60    3.5
## # … with 726 more rows, and 8 more variables: year_NA <fct>, latitude_NA <fct>,
## #   longitude_NA <fct>, sea_temp_c_NA <fct>, air_temp_c_NA <fct>,
## #   humidity_NA <fct>, wind_ew_NA <fct>, wind_ns_NA <fct>
```

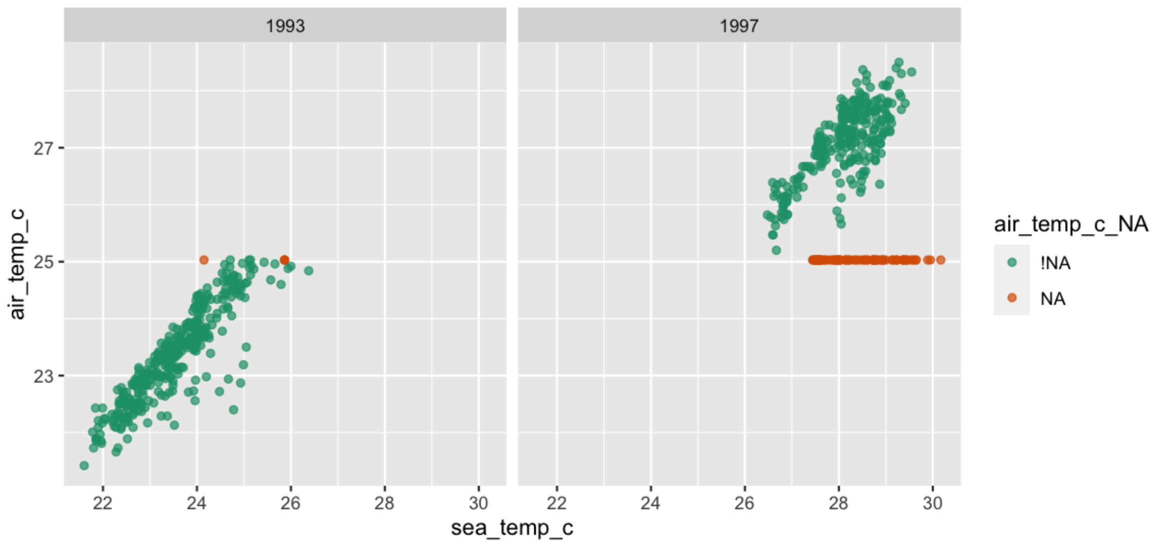
Imputing the Mean (ignoring year).

```
tao_imp_mean <- tao_shadow %>%
  mutate(sea_temp_c = impute_mean(sea_temp_c),
         air_temp_c = impute_mean(air_temp_c))

tao_shadow
## # A tibble: 736 x 16
##     year latitude longitude sea_temp_c air_temp_c humidity wind_ew wind_ns
##    <dbl>    <dbl>     <dbl>      <dbl>      <dbl>    <dbl>   <dbl>   <dbl>
##  1  1997        0      -110       27.6       27.1     79.6   -6.40    5.40
##  2  1997        0      -110       27.5       27.0     75.8   -5.30    5.30
##  3  1997        0      -110       27.6       27       76.5   -5.10    4.5
##  4  1997        0      -110       27.6       26.9     76.2   -4.90    2.5
##  5  1997        0      -110       27.6       26.8     76.4   -3.5     4.10
##  6  1997        0      -110       27.8       26.9     76.7   -4.40    1.60
##  7  1997        0      -110       28.0       27.0     76.5   -2       3.5
##  8  1997        0      -110       28.0       27.1     78.3   -3.70    4.5
##  9  1997        0      -110       28.0       27.2     78.6   -4.20    5
## 10  1997        0      -110       28.0       27.2     76.9   -3.60    3.5
## # … with 726 more rows, and 8 more variables: year_NA <fct>, latitude_NA <fct>,
## #   longitude_NA <fct>, sea_temp_c_NA <fct>, air_temp_c_NA <fct>,
```

Imputing the Mean (ignoring year).

```
ggplot(tao_imp_mean,
       aes(x = sea_temp_c,
           y = air_temp_c,
           colour = air_temp_c_N
  geom_point(alpha = 0.7) +
  facet_wrap(~year) +
  scale_colour_brewer(palette =
  theme(aspect.ratio = 1)
```

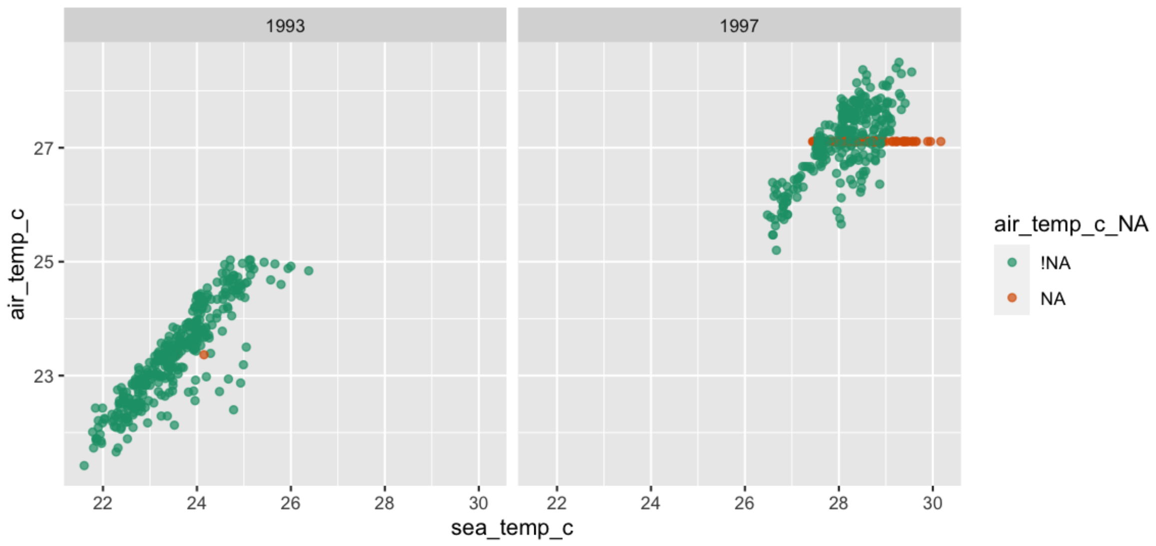


Impute Mean by year

```
tao_shadow <- tao_shadow %>%
  group_by(year) %>%
  mutate(sea_temp_c = impute_mean(sea_temp_c),
         air_temp_c = impute_mean(air_temp_c))
```

by year

```
ggplot(tao_shadow,
       aes(x = sea_temp_c,
           y = air_temp_c,
           colour=air_temp_c_NA)
  geom_point(alpha=0.7) +
  facet_wrap(~year) +
  scale_colour_brewer(palette="[
  theme(aspect.ratio=1)
```



Your Turn:

- lab quiz open (requires answering questions from Lab exercise)
- go to rstudio.cloud and finish final exercise

Resources

- [R-miss-Tastic]
- [naniar]
- [visdat]