

Modern R with tidyverse [Solutions]

Jonathan de Bruin & Barbara Vreede

Contents

Introduction	1
Technical requirements	1
1. Read and save data	2
2. Data visualisation	6
3. Data transformation	22

Introduction

In this document, we explore Crane migration, through the GPS data from 39 tagged individuals during the fall migration of 2017. The data was kindly provided for this course by Sasha Pekarsky at the Department of Ecology, Evolution, and Behavior at HUJI. The dataset has been modified by the course instructors.

Technical requirements

*This document is part of the workshop **Introduction to R & Data** by Utrecht University RDM Support.*

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2
## -- Attaching packages ----- tidyverse 1.2.1 --
## √ ggplot2 2.2.1      √ purrr   0.2.5
## √ tibble  1.4.2      √ dplyr    0.7.4
## √ tidyr   0.8.1      √ stringr 1.2.0
## √ readr   1.1.1      √forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.4.3
## Warning: package 'tidyr' was built under R version 3.4.4
## Warning: package 'purrr' was built under R version 3.4.4
## Warning: package 'dplyr' was built under R version 3.4.2
## Warning: package 'forcats' was built under R version 3.4.3

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()
```

1. Read and save data

Basic exercise I - Read data into R

a) Find the delimiter of the dataset HUJI_Crane_Israel_GPRS.csv

Use the data import tool of RStudio or use a text editor to inspect the document. The delimiter is ,.

b) Read the data HUJI_Crane_Israel_GPRS.csv into R

```
data_crane <- read_csv('data/HUJI_Crane_Israel_GPRS.csv')

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   visible = col_character(),
##   timestamp = col_datetime(format = ""),
##   battery_charge_percent = col_integer(),
##   eobs_activity = col_character(),
##   eobs_activity_samples = col_character(),
##   eobs_battery_voltage = col_integer(),
##   eobs_fix_battery_voltage = col_integer(),
##   eobs_start_timestamp = col_character(),
##   eobs_status = col_character(),
##   eobs_temperature = col_integer(),
##   eobs_type_of_fix = col_integer(),
##   eobs_used_time_to_get_fix = col_integer(),
##   gps_satellite_count = col_integer(),
##   import_marked_outlier = col_character(),
##   orn_transmission_protocol = col_character(),
##   sensor_type = col_character(),
##   individual_taxon_canonical_name = col_character(),
##   individual_local_identifier = col_character(),
##   study_name = col_character()
## )

## See spec(...) for full column specifications.

head(data_crane)

## # A tibble: 6 x 42
##   event_id visible timestamp           location_long location_lat
##   <dbl>    <chr>   <dttm>                <dbl>        <dbl>
## 1 3.79e9  true    2017-10-15 00:00:16     40.3       55.3
## 2 3.80e9  true    2017-10-15 00:00:16     40.3       55.3
## 3 3.79e9  true    2017-10-15 02:00:23     40.3       55.3
## 4 3.80e9  true    2017-10-15 02:00:23     40.3       55.3
## 5 3.79e9  true    2017-10-15 04:00:23     40.3       55.3
## 6 3.80e9  true    2017-10-15 04:00:23     40.3       55.3
## # ... with 37 more variables: acceleration_raw_x <dbl>,
## #   acceleration_raw_y <dbl>, acceleration_raw_z <dbl>,
## #   bar_barometric_height <dbl>, battery_charge_percent <int>,
## #   battery_charging_current <dbl>, eobs_activity <chr>,
## #   eobs_activity_samples <chr>, eobs_battery_voltage <int>,
## #   eobs_fix_battery_voltage <int>,
```

```

## #   eobs_horizontal_accuracy_estimate <dbl>, eobs_key_bin_checksum <dbl>,
## #   eobs_speed_accuracy_estimate <dbl>, eobs_start_timestamp <chr>,
## #   eobs_status <chr>, eobs_temperature <int>, eobs_type_of_fix <int>,
## #   eobs_used_time_to_get_fix <int>, external_temperature <dbl>,
## #   gps_hdop <dbl>, gps_satellite_count <int>, gps_time_to_fix <dbl>,
## #   ground_speed <dbl>, heading <dbl>, height_above_ellipsoid <dbl>,
## #   height_above_msl <dbl>, import_marked_outlier <chr>,
## #   gls_light_level <dbl>, mag_magnetic_field_raw_x <dbl>,
## #   mag_magnetic_field_raw_y <dbl>, mag_magnetic_field_raw_z <dbl>,
## #   orn_transmission_protocol <chr>, tag_voltage <dbl>, sensor_type <chr>,
## #   individual_taxon_canonical_name <chr>,
## #   individual_local_identifier <chr>, study_name <chr>

```

c) Load `readxl` (to read Excel files)

```

library(readxl)

## Warning: package 'readxl' was built under R version 3.4.4
# Hint: type ?read_excel in your console

```

c) Read the additional observations `crane_additional_observations.xlsx`.

```

data_crane_additional <- read_excel('data/crane_additional_observations.xlsx')

head(data_crane_additional)

## # A tibble: 6 x 4
##       event_id meas_1 meas_2 meas_3
##       <dbl>    <dbl>    <dbl>    <dbl>
## 1 3794510958  1.15    1.93    0.503
## 2 3796034354  0.376   0.307   0.341
## 3 3794510962  0.217   0.829   0.963
## 4 3796034355  0.684   0.250   0.129
## 5 3794510961  1.30    1.87    1.14
## 6 3796034356  0.847   5.14    0.964

```

Basic exercise II - Dataset properties

```

glimpse(data_crane)

## Observations: 20,873
## Variables: 42
## $ event_id                  <dbl> 3794510958, 3796034354, 3794...
## $ visible                    <chr> "true", "true", "true", "tru...
## $ timestamp                  <dttm> 2017-10-15 00:00:16, 2017-1...
## $ location_long               <dbl> 40.31509, 40.31509, 40.31481...
## $ location_lat                <dbl> 55.34764, 55.34764, 55.34792...
## $ acceleration_raw_x          <dbl> NA, NA, NA, NA, NA, NA, ...
## $ acceleration_raw_y          <dbl> NA, NA, NA, NA, NA, NA, ...
## $ acceleration_raw_z          <dbl> NA, NA, NA, NA, NA, NA, ...
## $ bar_barometric_height       <dbl> NA, NA, NA, NA, NA, NA, ...
## $ battery_charge_percent      <int> NA, NA, NA, NA, NA, NA, ...

```

```

## $ battery_charging_current          <dbl> NA, NA, NA, NA, NA, NA, ...
## $ eobs_activity                   <chr> NA, NA, NA, NA, NA, NA, ...
## $ eobs_activity_samples            <chr> NA, NA, NA, NA, NA, NA, ...
## $ eobs_battery_voltage             <int> 3784, 3818, 3784, 3818, 3784...
## $ eobs_fix_battery_voltage        <int> NA, 3801, NA, 3798, NA, 3796...
## $ eobs_horizontal_accuracy_estimate <dbl> NA, 48.90, NA, 2.56, NA, 3.8...
## $ eobs_key_bin_checksum           <dbl> 0, 2787211109, 0, 1067160123...
## $ eobs_speed_accuracy_estimate    <dbl> NA, 1.54, NA, 0.33, NA, 0.34...
## $ eobs_start_timestamp            <chr> NA, "2017,15+Oct", NA, "2017...
## $ eobs_status                     <chr> NA, "A", NA, "A", NA, "A", N...
## $ eobs_temperature                <int> NA, -7, NA, -7, NA, -8, NA, ...
## $ eobs_type_of_fix                <int> NA, 3, NA, 3, NA, 3, NA, 3, ...
## $ eobs_used_time_to_get_fix      <int> NA, 15, NA, 22, NA, 21, NA, ...
## $ external_temperature             <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ gps_hdop                        <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ gps_satellite_count              <int> NA, NA, NA, NA, NA, NA, NA, ...
## $ gps_time_to_fix                 <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ ground_speed                    <dbl> NA, 0.34, NA, 0.01, NA, 12.8...
## $ heading                          <dbl> NA, 340.13, NA, 0.00, NA, 22...
## $ height_above_ellipsoid          <dbl> NA, 127.0, NA, 112.2, NA, 20...
## $ height_above_msl                <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ import_marked_outlier           <chr> NA, NA, NA, NA, NA, NA, NA, ...
## $ gls_light_level                  <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ mag_magnetic_field_raw_x        <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ mag_magnetic_field_raw_y        <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ mag_magnetic_field_raw_z        <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ orn_transmission_protocol       <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ tag_voltage                      <dbl> NA, NA, NA, NA, NA, NA, NA, ...
## $ sensor_type                     <chr> "gps", "gps", "gps", "gps", ...
## $ individual_taxon_canonical_name <chr> "Grus grus", "Grus grus", "G...
## $ individual_local_identifier     <chr> "L6037", "L6037", "L6037", "...
## $ study_name                       <chr> "Huji JNF Crane israel GPRS"...

glimpse(data_crane_additional)

```

```

## Observations: 20,873
## Variables: 4
## $ event_id <dbl> 3794510958, 3796034354, 3794510962, 3796034355, 37945...
## $ meas_1   <dbl> 1.1485201, 0.3756848, 0.2170456, 0.6836040, 1.2951363...
## $ meas_2   <dbl> 1.9269063, 0.3073029, 0.8286311, 0.2501690, 1.8683569...
## $ meas_3   <dbl> 0.50257465, 0.34084276, 0.96295022, 0.12898235, 1.136...

```

Reading exercise - `readr` versus base R

Optional exercise (+) - Save data to a CSV file with delimiter ;.

```

# create a directory for the output file
if (!dir.exists('tmp')){
  dir.create("tmp")
}

write_delim(data_crane, 'tmp/data_crane_csv_file.csv', delim = ';')

```

Optional exercise (++) - Read SPSS, SAS, Excel and STATA data files

```
library("haven") # to read and write SPSS, STATA and SAS files
```

```
## Warning: package 'haven' was built under R version 3.4.4
```

a) Write tibble data_crane to SPSS, SAS, STATA data files.

```
# create a directory
if (!dir.exists('tmp')){
  dir.create("tmp")
}

# read and write files
write_sav(data_crane, file.path("tmp", "crane_spss.sav"))
read_sav(file.path("tmp", "crane_spss.sav"))

## # A tibble: 20,873 x 42
##   event_id visible timestamp           location_long location_lat
##   <dbl>     <chr>    <dttm>                <dbl>        <dbl>
## 1 3.79e9  true  2017-10-15 00:00:16      40.3       55.3
## 2 3.80e9  true  2017-10-15 00:00:16      40.3       55.3
## 3 3.79e9  true  2017-10-15 02:00:23      40.3       55.3
## 4 3.80e9  true  2017-10-15 02:00:23      40.3       55.3
## 5 3.79e9  true  2017-10-15 04:00:23      40.3       55.3
## 6 3.80e9  true  2017-10-15 04:00:23      40.3       55.3
## 7 3.79e9  true  2017-10-15 05:00:14      40.1       55.3
## 8 3.80e9  true  2017-10-15 05:00:14      40.1       55.3
## 9 3.80e9  true  2017-10-15 06:00:23      40.1       55.3
## 10 3.80e9 true  2017-10-15 08:00:18      40.1       55.3
## # ... with 20,863 more rows, and 37 more variables:
## #   acceleration_raw_x <dbl>, acceleration_raw_y <dbl>,
## #   acceleration_raw_z <dbl>, bar_barometric_height <dbl>,
## #   battery_charge_percent <dbl>, battery_charging_current <dbl>,
## #   eobs_activity <chr>, eobs_activity_samples <chr>,
## #   eobs_battery_voltage <dbl>, eobs_fix_battery_voltage <dbl>,
## #   eobs_horizontal_accuracy_estimate <dbl>, eobs_key_bin_checksum <dbl>,
## #   eobs_speed_accuracy_estimate <dbl>, eobs_start_timestamp <chr>,
## #   eobs_status <chr>, eobs_temperature <dbl>, eobs_type_of_fix <dbl>,
## #   eobs_used_time_to_get_fix <dbl>, external_temperature <dbl>,
## #   gps_hdop <dbl>, gps_satellite_count <dbl>, gps_time_to_fix <dbl>,
## #   ground_speed <dbl>, heading <dbl>, height_above_ellipsoid <dbl>,
## #   height_above_msl <dbl>, import_marked_outlier <chr>,
## #   gls_light_level <dbl>, mag_magnetic_field_raw_x <dbl>,
## #   mag_magnetic_field_raw_y <dbl>, mag_magnetic_field_raw_z <dbl>,
## #   orn_transmission_protocol <chr>, tag_voltage <dbl>, sensor_type <chr>,
## #   individual_taxon_canonical_name <chr>,
## #   individual_local_identifier <chr>, study_name <chr>

# write_sas(data_crane, file.path("tmp", "crane_sas.sas7bdat"))
# read_sas(file.path("tmp", "crane_sas.sas7bdat"))

# write_dta(data_crane, file.path("tmp", "crane_stata.dta"))
# read_dta(file.path("tmp", "crane_stata.dta"))
```

b) Write tibble `data_crane` to Excel.

This is not possible with `tidyverse` at the moment. `readxl` only support Excel file reading. This is not a problem for a researcher, because we don't use Excel, do we?

Optional exercise (++) - Parse datetime columns

```
read_csv('data/HUJI_Crane_Israel_GPRS.csv',
  col_types = cols(
    . = col_guess(),
    eobs_start_timestamp = col_date(format="%Y,%d+%b")
  ))

## Warning: The following named parsers don't match the column names: .

## # A tibble: 20,873 x 42
##   event_id visible timestamp           location_long location_lat
##   <dbl>     <chr>   <dttm>            <dbl>        <dbl>
## 1 3.79e9  true  2017-10-15 00:00:16      40.3       55.3
## 2 3.80e9  true  2017-10-15 00:00:16      40.3       55.3
## 3 3.79e9  true  2017-10-15 02:00:23      40.3       55.3
## 4 3.80e9  true  2017-10-15 02:00:23      40.3       55.3
## 5 3.79e9  true  2017-10-15 04:00:23      40.3       55.3
## 6 3.80e9  true  2017-10-15 04:00:23      40.3       55.3
## 7 3.79e9  true  2017-10-15 05:00:14      40.1       55.3
## 8 3.80e9  true  2017-10-15 05:00:14      40.1       55.3
## 9 3.80e9  true  2017-10-15 06:00:23      40.1       55.3
## 10 3.80e9 true  2017-10-15 08:00:18       40.1       55.3
## # ... with 20,863 more rows, and 37 more variables:
## #   acceleration_raw_x <dbl>, acceleration_raw_y <dbl>,
## #   acceleration_raw_z <dbl>, bar_barometric_height <dbl>,
## #   battery_charge_percent <int>, battery_charging_current <dbl>,
## #   eobs_activity <chr>, eobs_activity_samples <chr>,
## #   eobs_battery_voltage <int>, eobs_fix_battery_voltage <int>,
## #   eobs_horizontal_accuracy_estimate <dbl>, eobs_key_bin_checksum <dbl>,
## #   eobs_speed_accuracy_estimate <dbl>, eobs_start_timestamp <date>,
## #   eobs_status <chr>, eobs_temperature <int>, eobs_type_of_fix <int>,
## #   eobs_used_time_to_get_fix <int>, external_temperature <dbl>,
## #   gps_hdop <dbl>, gps_satellite_count <int>, gps_time_to_fix <dbl>,
## #   ground_speed <dbl>, heading <dbl>, height_above_ellipsoid <dbl>,
## #   height_above_msl <dbl>, import_marked_outlier <chr>,
## #   gls_light_level <dbl>, mag_magnetic_field_raw_x <dbl>,
## #   mag_magnetic_field_raw_y <dbl>, mag_magnetic_field_raw_z <dbl>,
## #   orn_transmission_protocol <chr>, tag_voltage <dbl>, sensor_type <chr>,
## #   individual_taxon_canonical_name <chr>,
## #   individual_local_identifier <chr>, study_name <chr>
```

2. Data visualisation

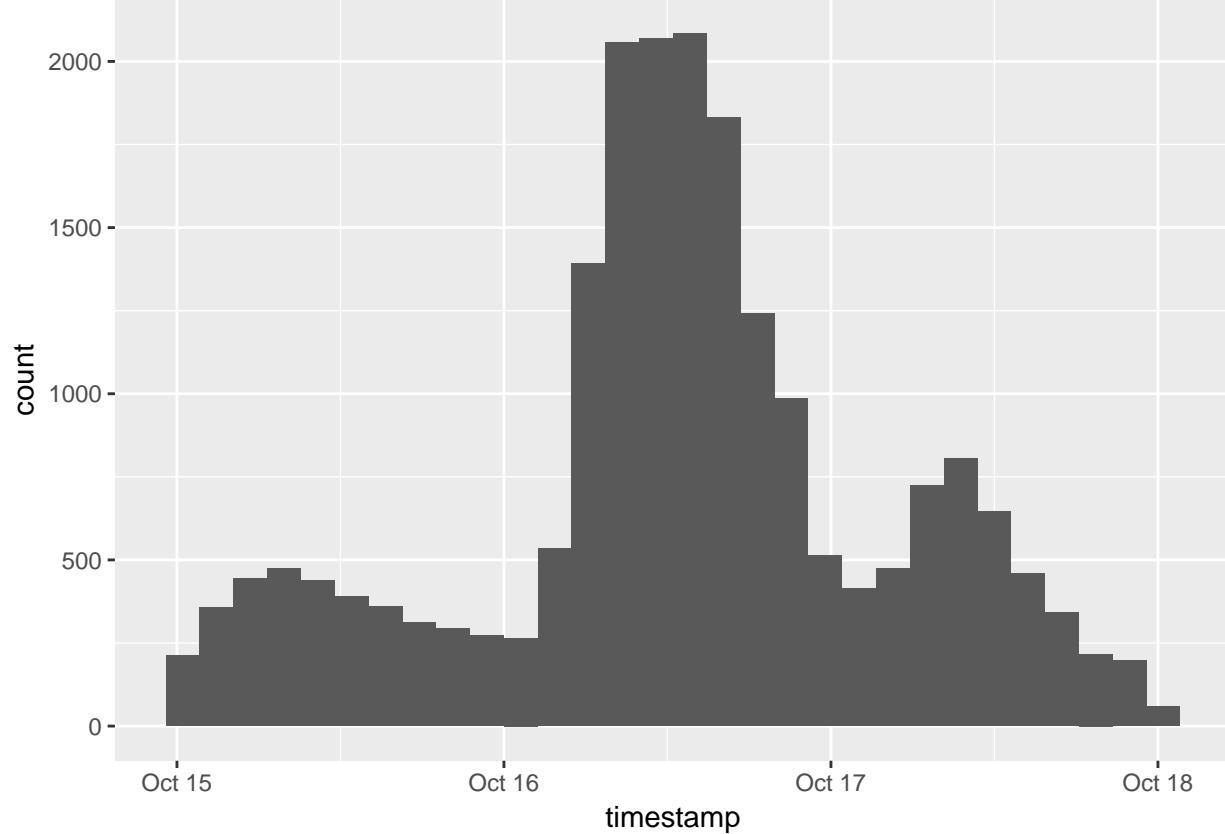
Basic exercise I - Quick plots of the `data_crane`.

a) Single column plots

Make a quick plot of at least 5 columns in the `data_crane` dataset.

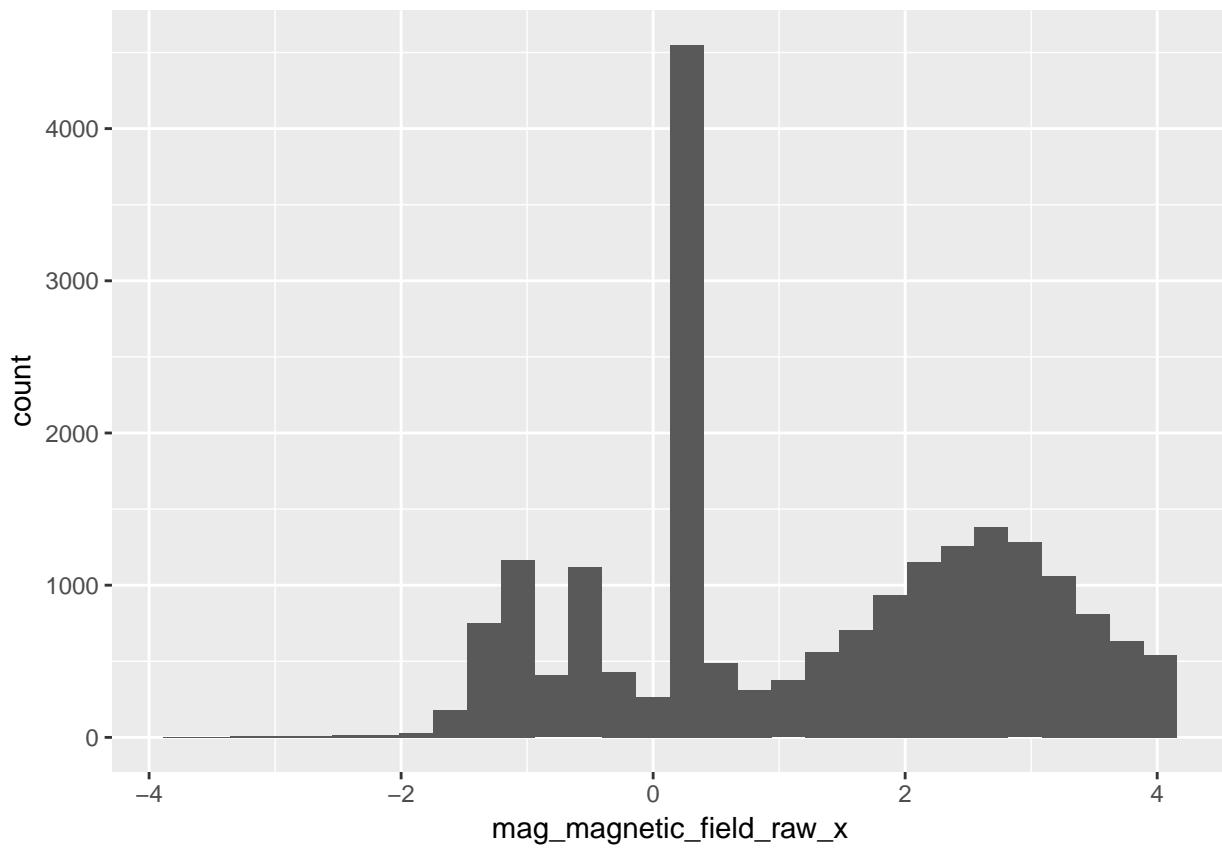
```
qplot(timestamp, data=data_crane)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

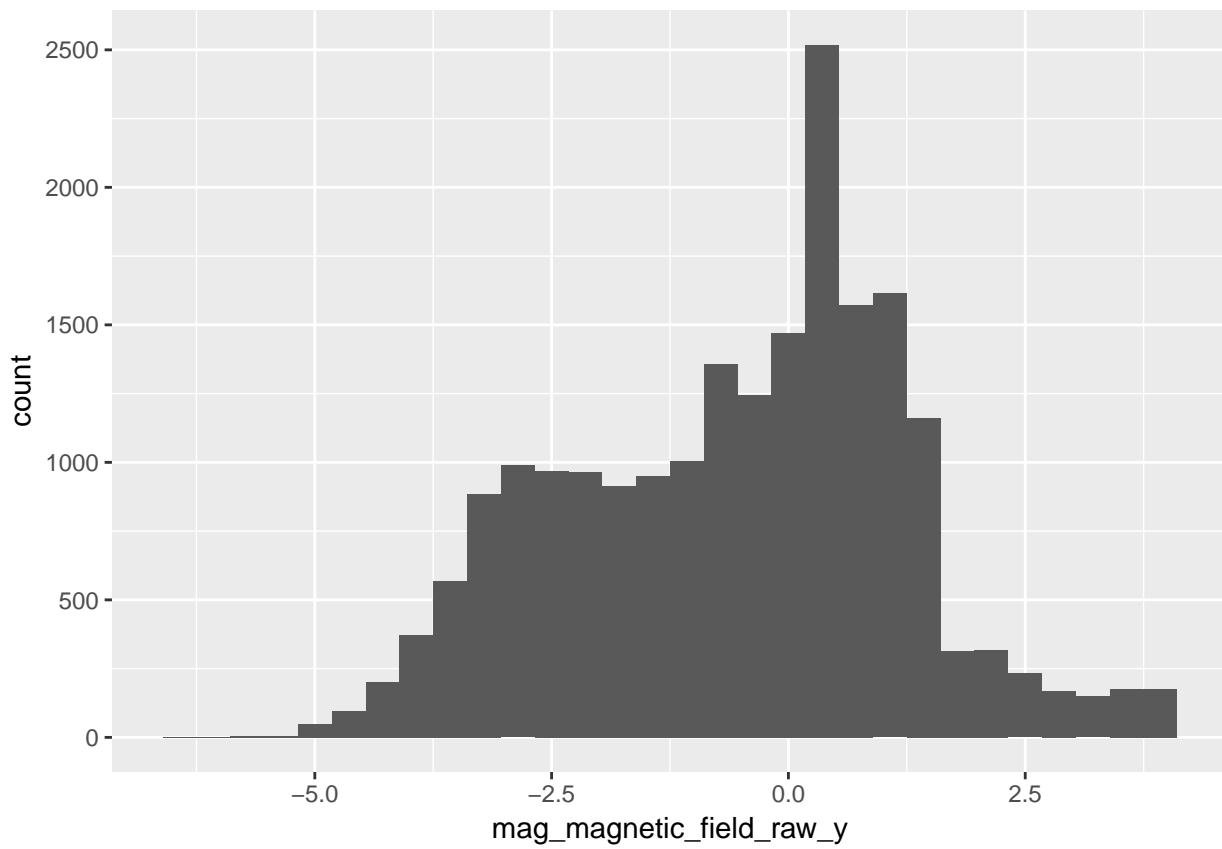


```
qplot(mag_magnetic_field_raw_x, data=data_crane)
```

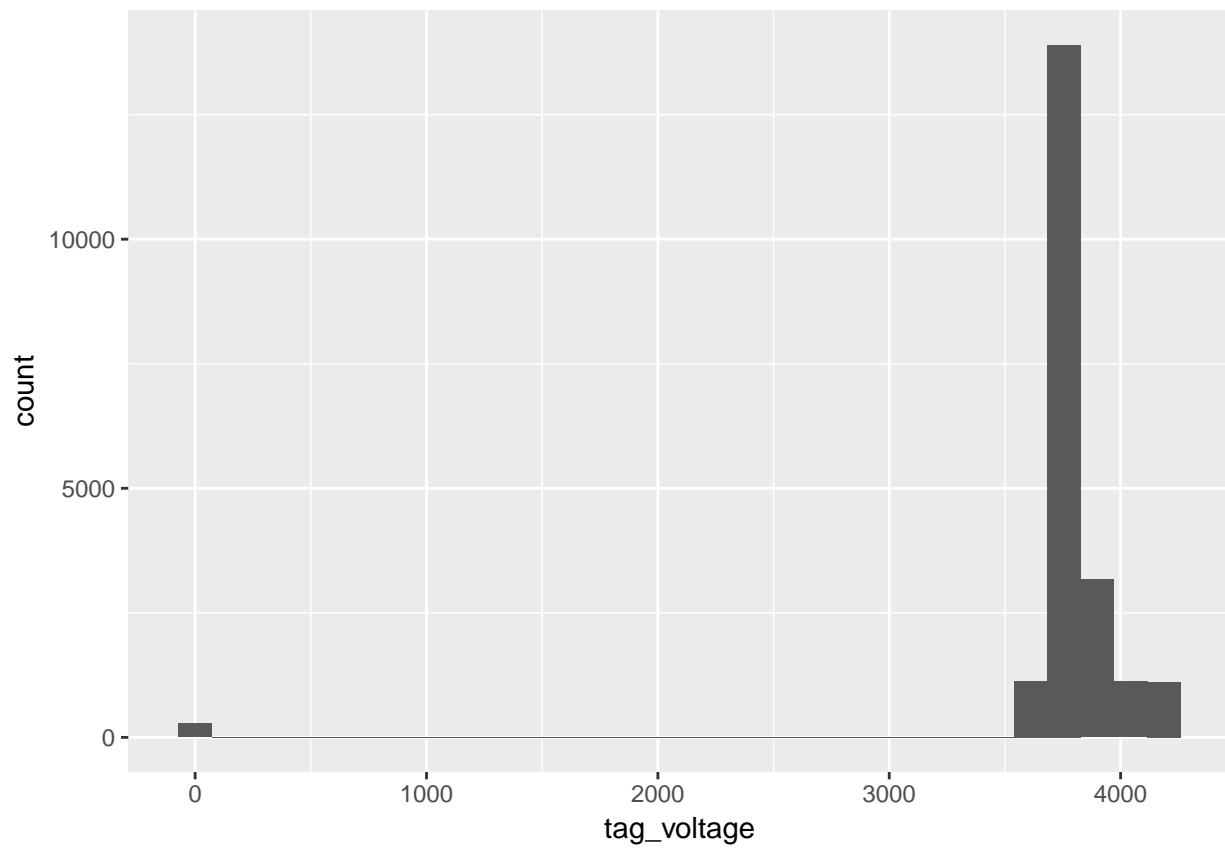
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 432 rows containing non-finite values (stat_bin).
```



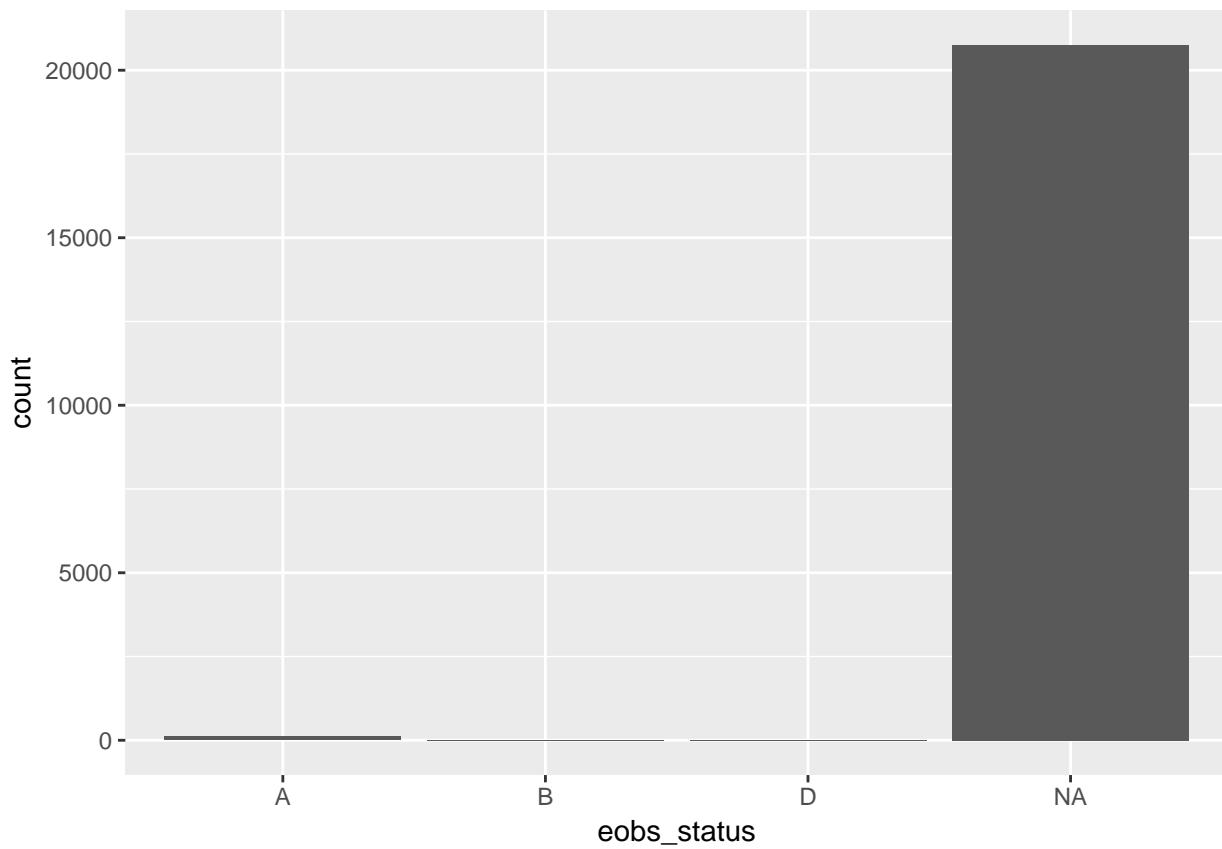
```
qplot(mag_magnetic_field_raw_y, data=data_crane)  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## Warning: Removed 432 rows containing non-finite values (stat_bin).
```



```
qplot(tag_voltage, data=data_crane)  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## Warning: Removed 155 rows containing non-finite values (stat_bin).
```

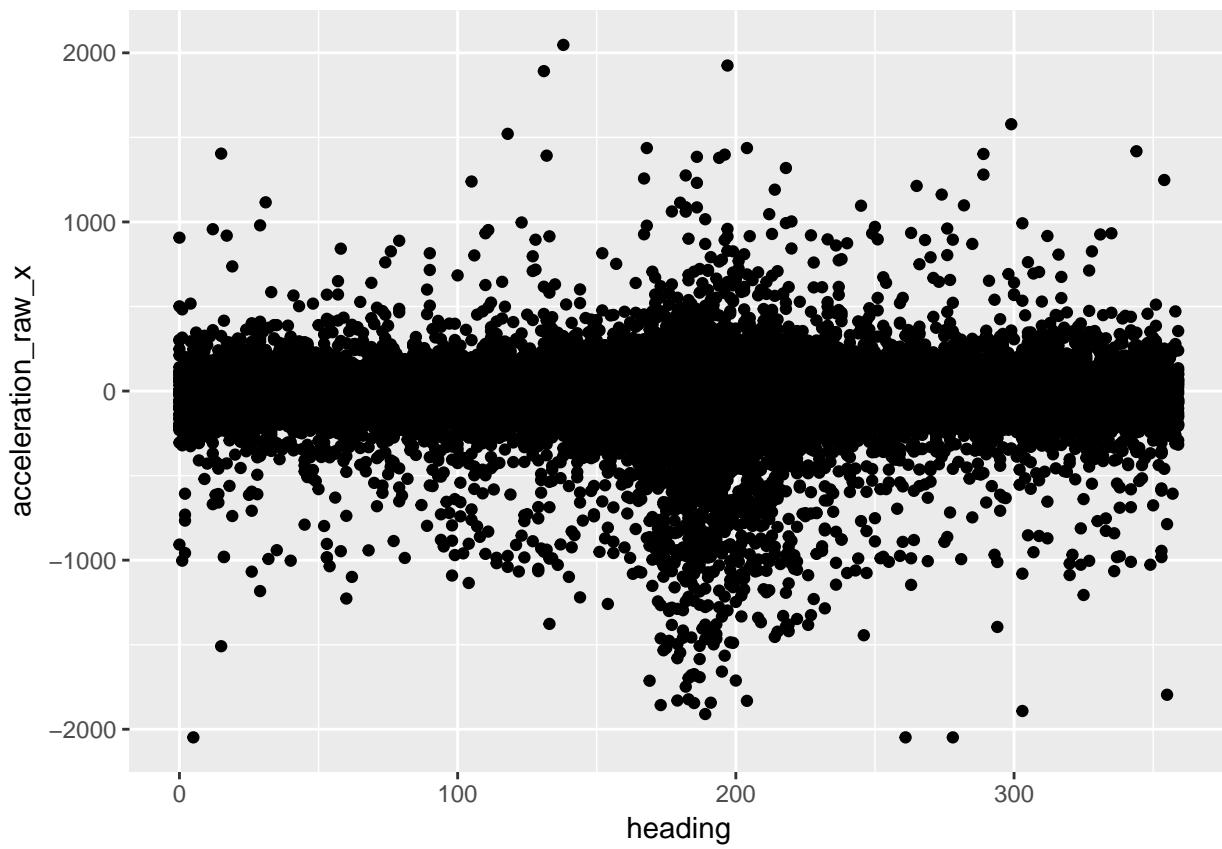


```
qplot(eobs_status, data=data_crane)
```



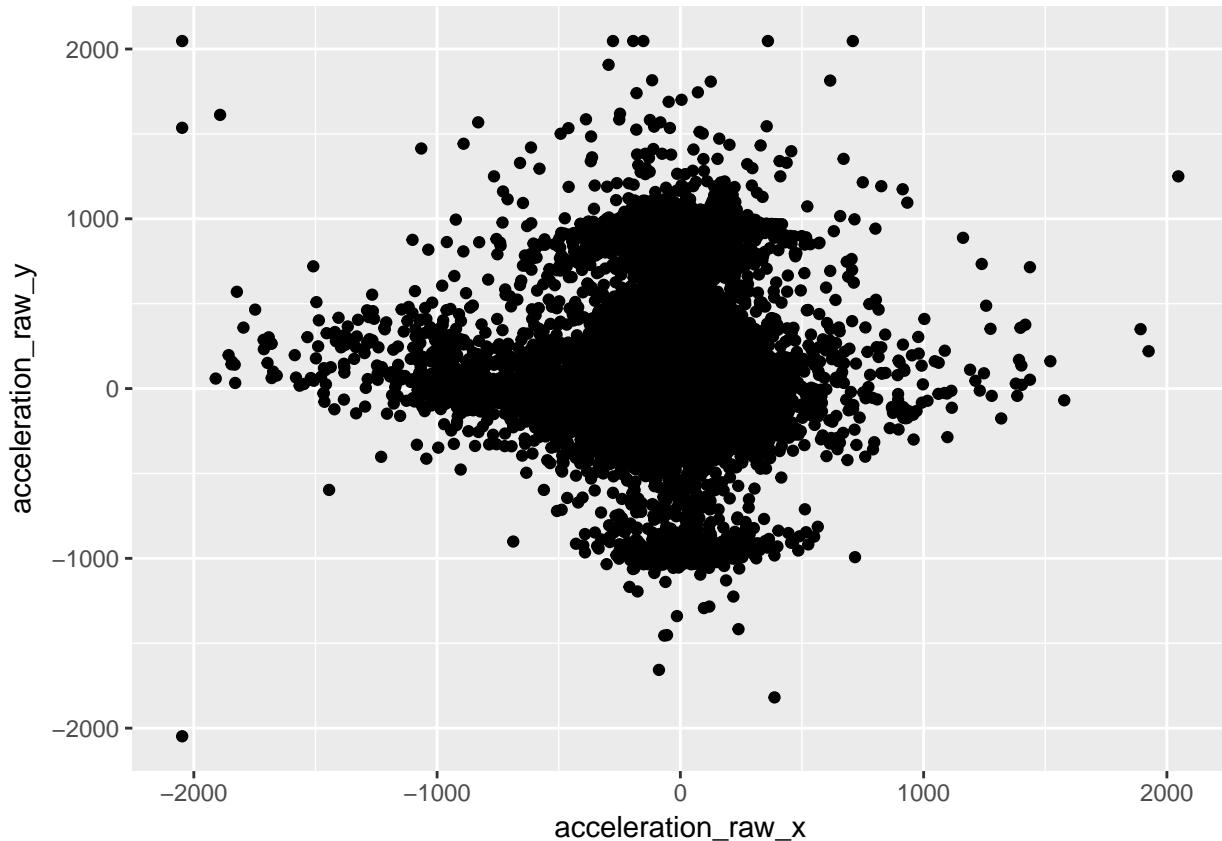
b) Two column in plots

```
qplot(heading, acceleration_raw_x, data=data_crane)  
## Warning: Removed 432 rows containing missing values (geom_point).
```



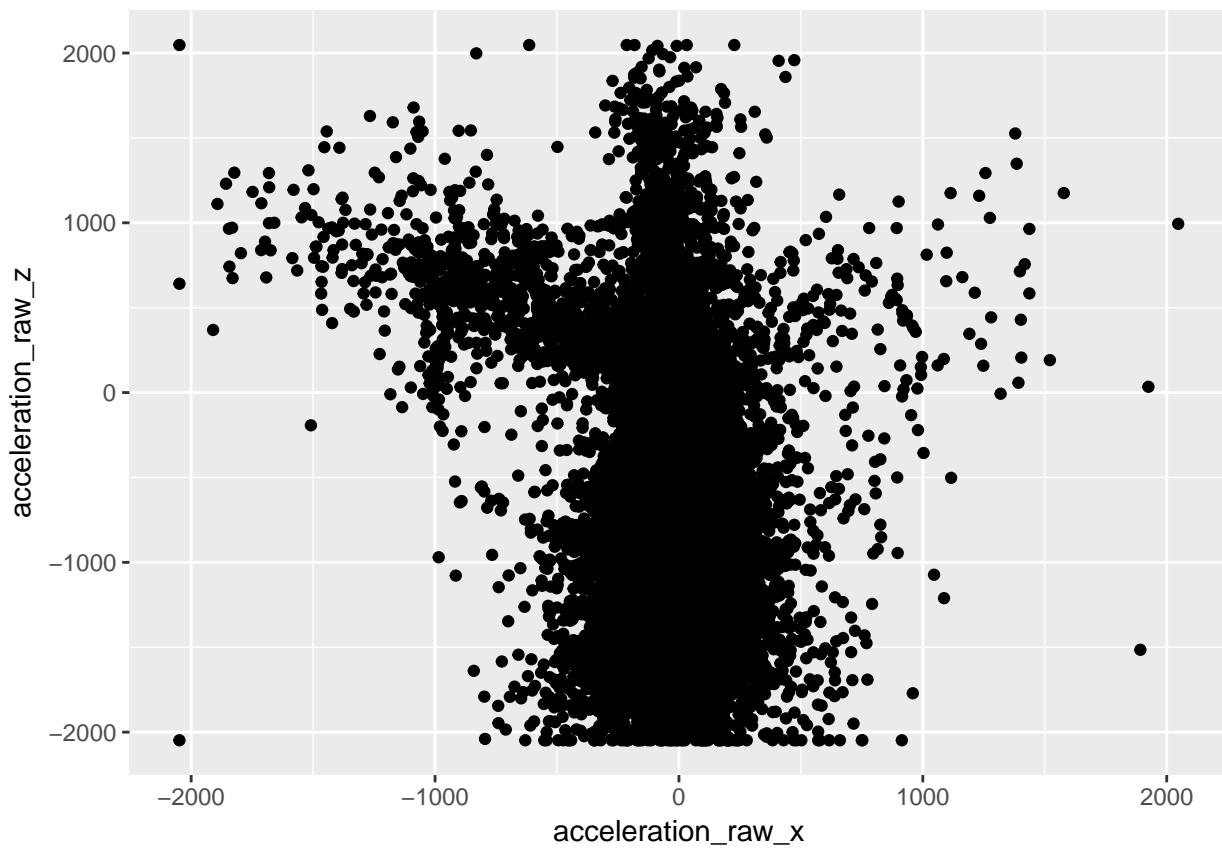
```
qplot(acceleration_raw_x, acceleration_raw_y, data=data_crane)
```

```
## Warning: Removed 432 rows containing missing values (geom_point).
```



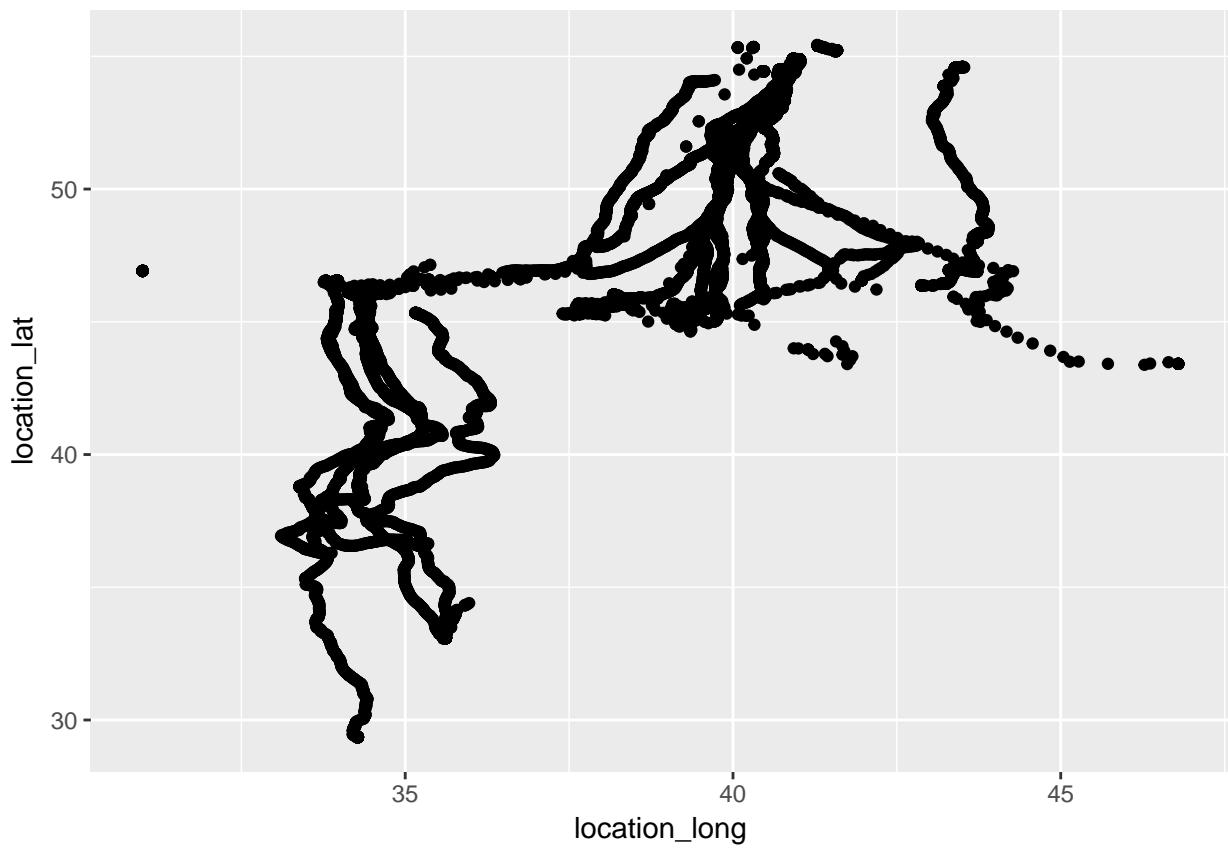
```
qplot(acceleration_raw_x, acceleration_raw_y, data=data_crane)
```

```
## Warning: Removed 432 rows containing missing values (geom_point).
```



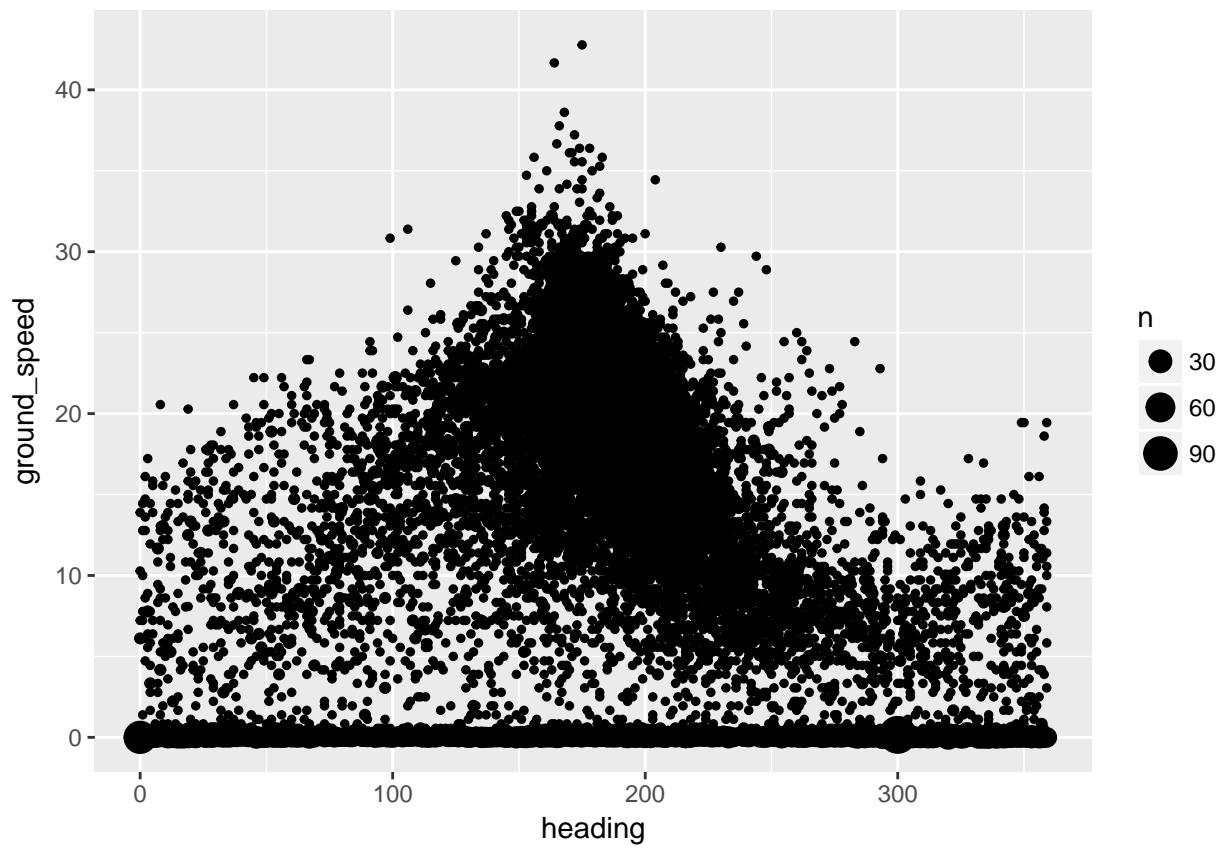
```
qplot(location_long, location_lat, data=data_crane)
```

```
## Warning: Removed 200 rows containing missing values (geom_point).
```



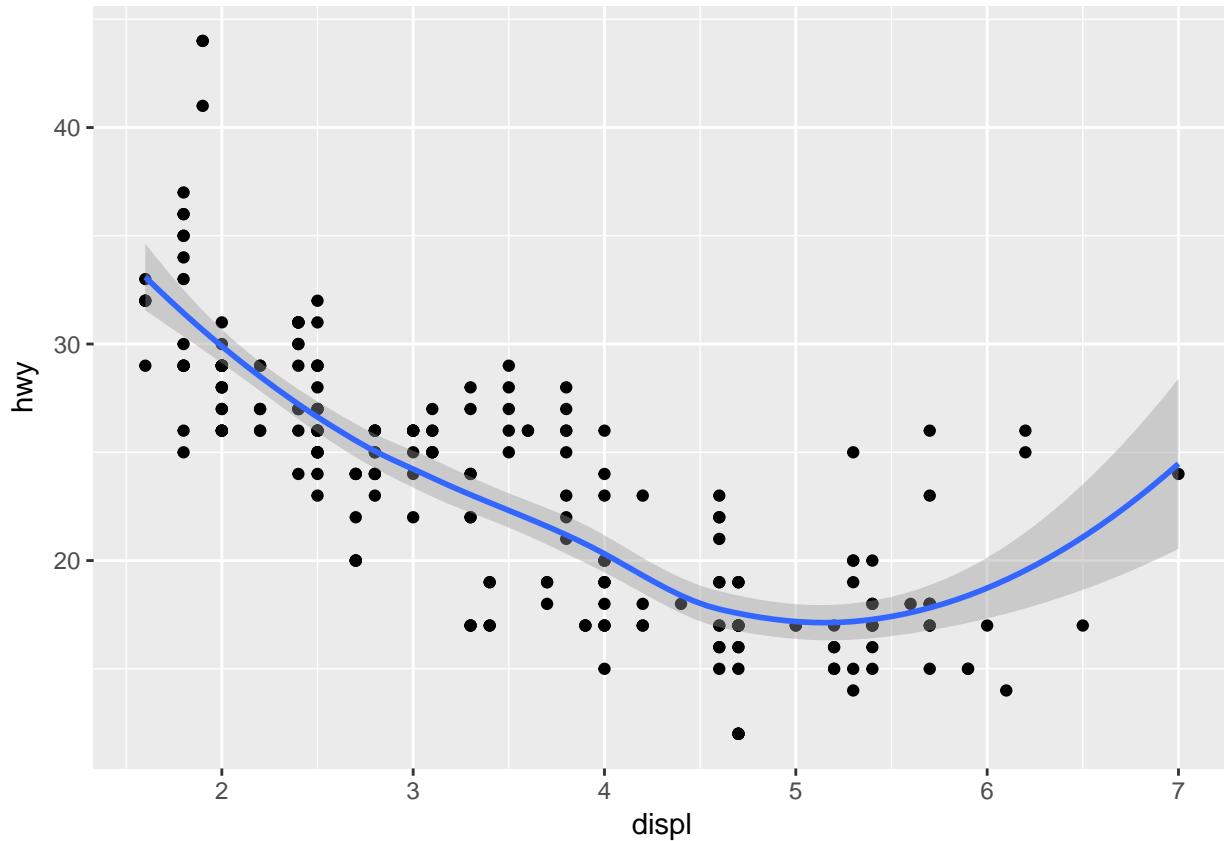
Basic exercise II - Using ggplot for plotting

```
ggplot(data_crane, aes(heading, ground_speed)) +  
  geom_count()  
  
## Warning: Removed 40 rows containing non-finite values (stat_sum).
```



Reading exercise - Statistical layers for graphs.

Statistical layers reveal a strong power of ggplot. The following graph illustrates this:



Read Chapter 3.6 of R for Data Science and experiment with the function `geom_smooth()` and `stat_smooth()`. What is the difference between both functions?

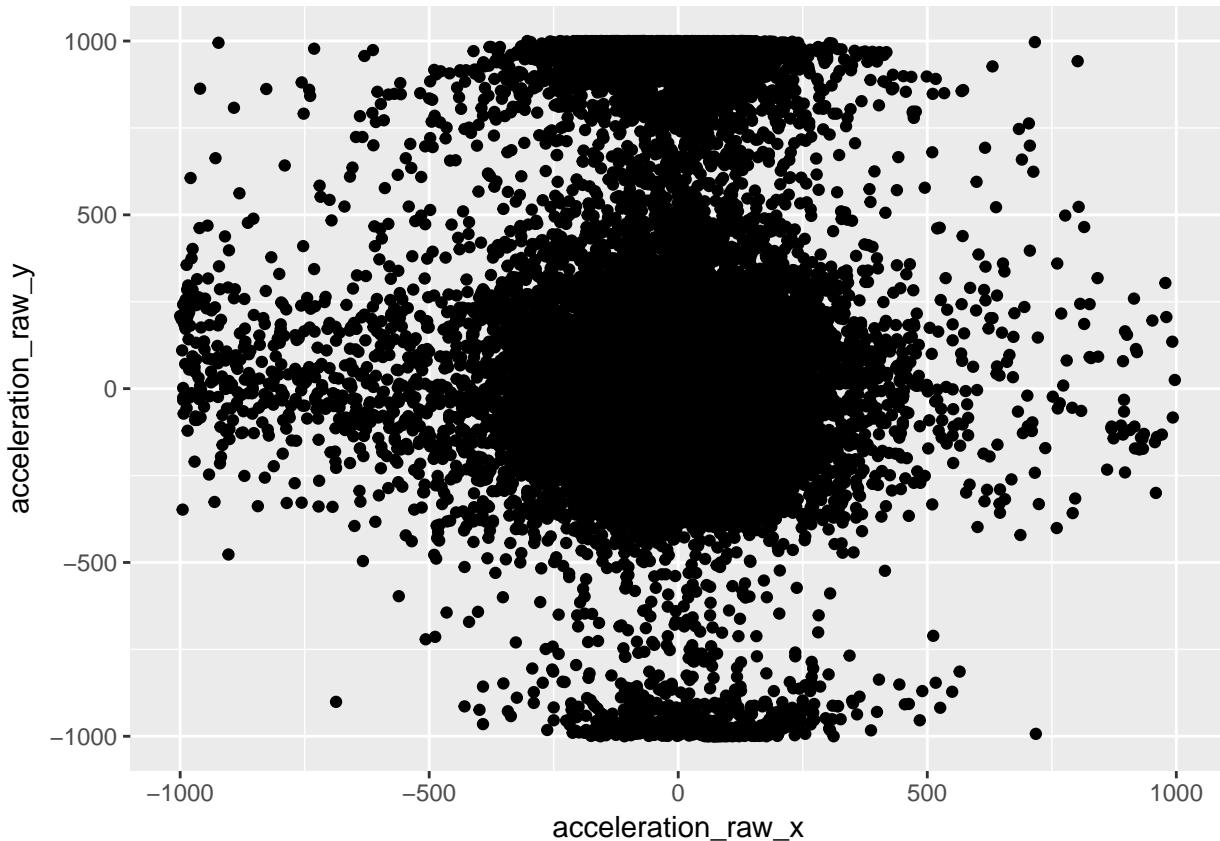
Optional exercise (+) - Scale axes

Scaling of the axes with a ggplot is easy. Take a look at the *data-visualization* cheat sheet. In the bottom right corner of the cheat sheet, you will find the code needed to scale the axes.

Zoom in on the acceleration between -1000 and 1000 for both the x-axis and y-axis.

```
ggplot(data_crane, aes(acceleration_raw_x, acceleration_raw_y)) +
  geom_point() +
  scale_x_continuous(limits = c(-1000, 1000)) +
  scale_y_continuous(limits = c(-1000, 1000))

## Warning: Removed 2436 rows containing missing values (geom_point).
```



Optional exercise (++) - Plot the crane positions on a map

a) Install package the maps

```
# install.packages('maps')
library(maps)

## Warning: package 'maps' was built under R version 3.4.4
##
## Attaching package: 'maps'
## The following object is masked from 'package:purrr':
## 
##     map
```

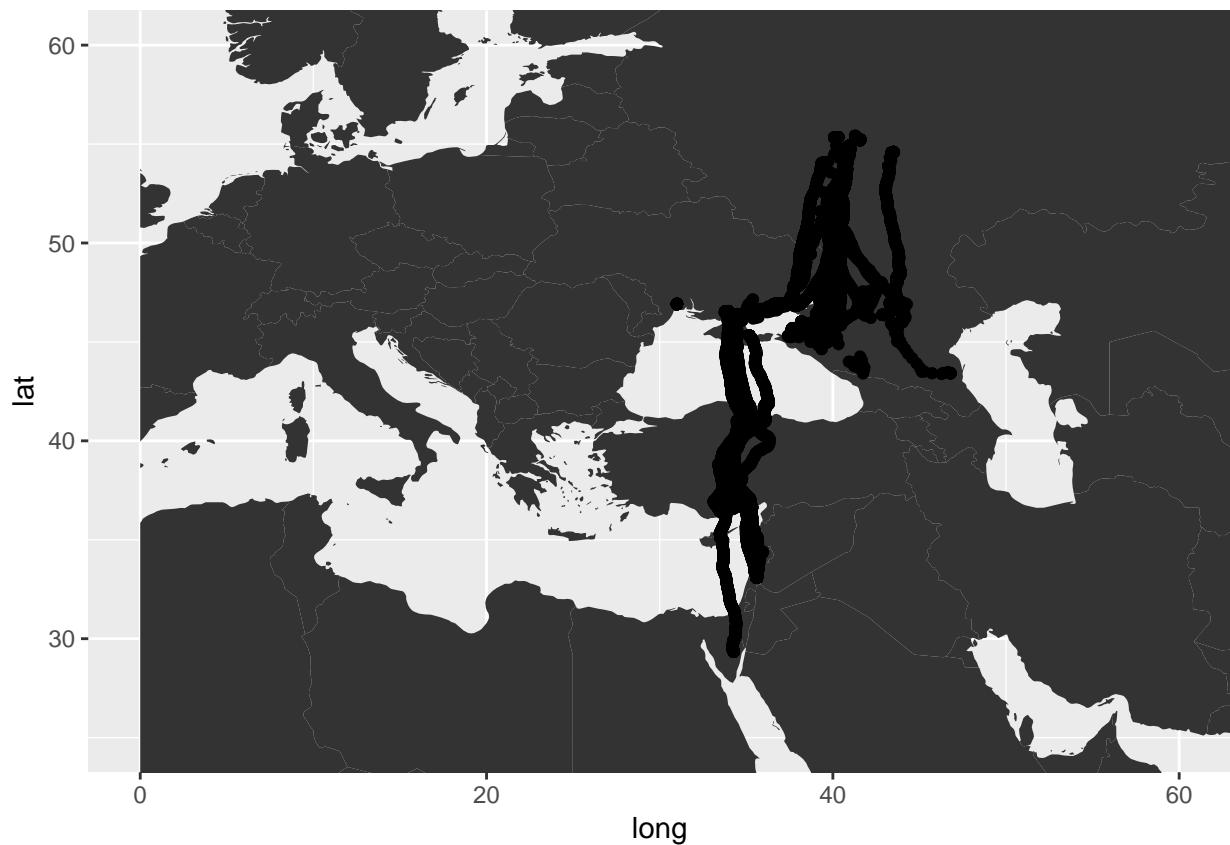
b) Plot the crane data on a map.

```
world_map_polygon <- map_data("world2")

ggplot(data_crane) +
  geom_map(data = world_map_polygon, map= world_map_polygon, aes(x=long, y = lat, map_id = region)) +
  scale_x_continuous(limits = c(0, 60)) +
  scale_y_continuous(limits = c(25, 60)) +
  geom_point(data = data_crane, aes(x = location_long, y = location_lat))

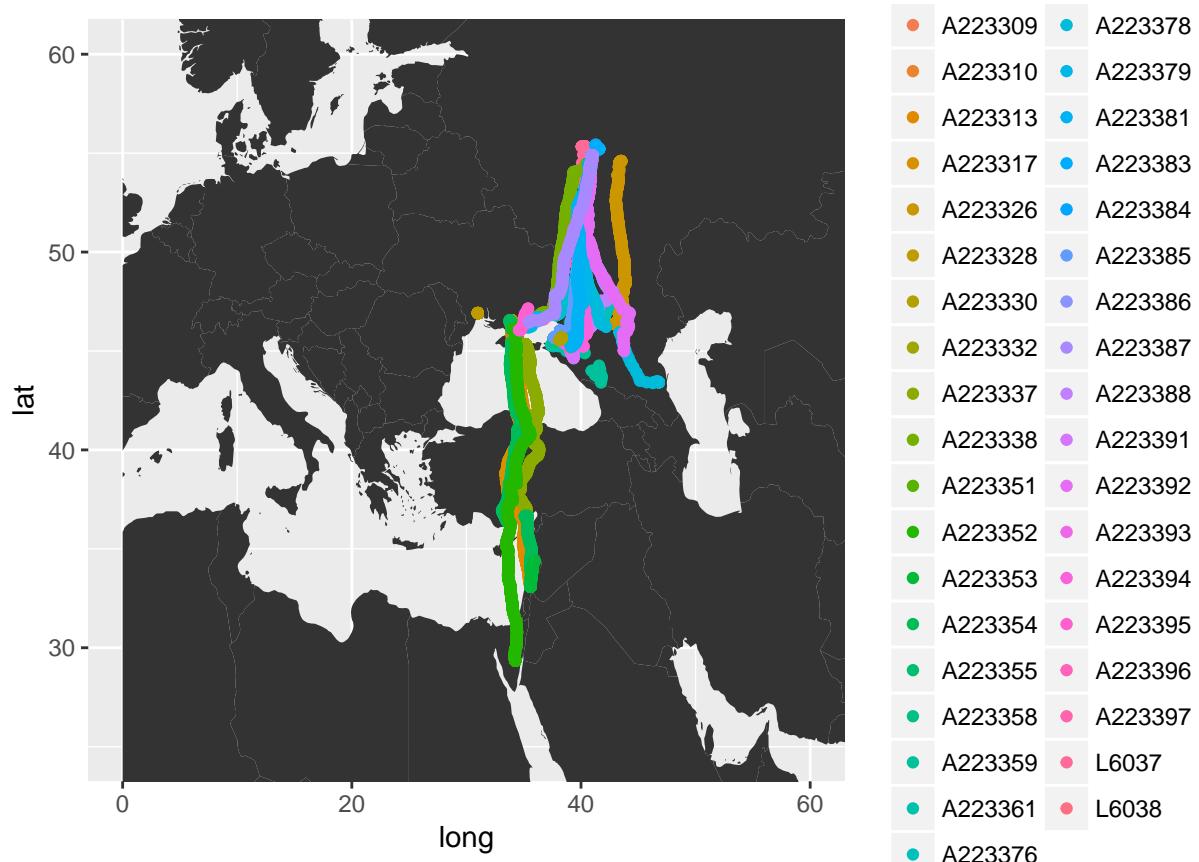
## Warning: Ignoring unknown aesthetics: x, y
```

```
## Warning: Removed 200 rows containing missing values (geom_point).
```



c) Use an individual identifier to colour the different cranes.

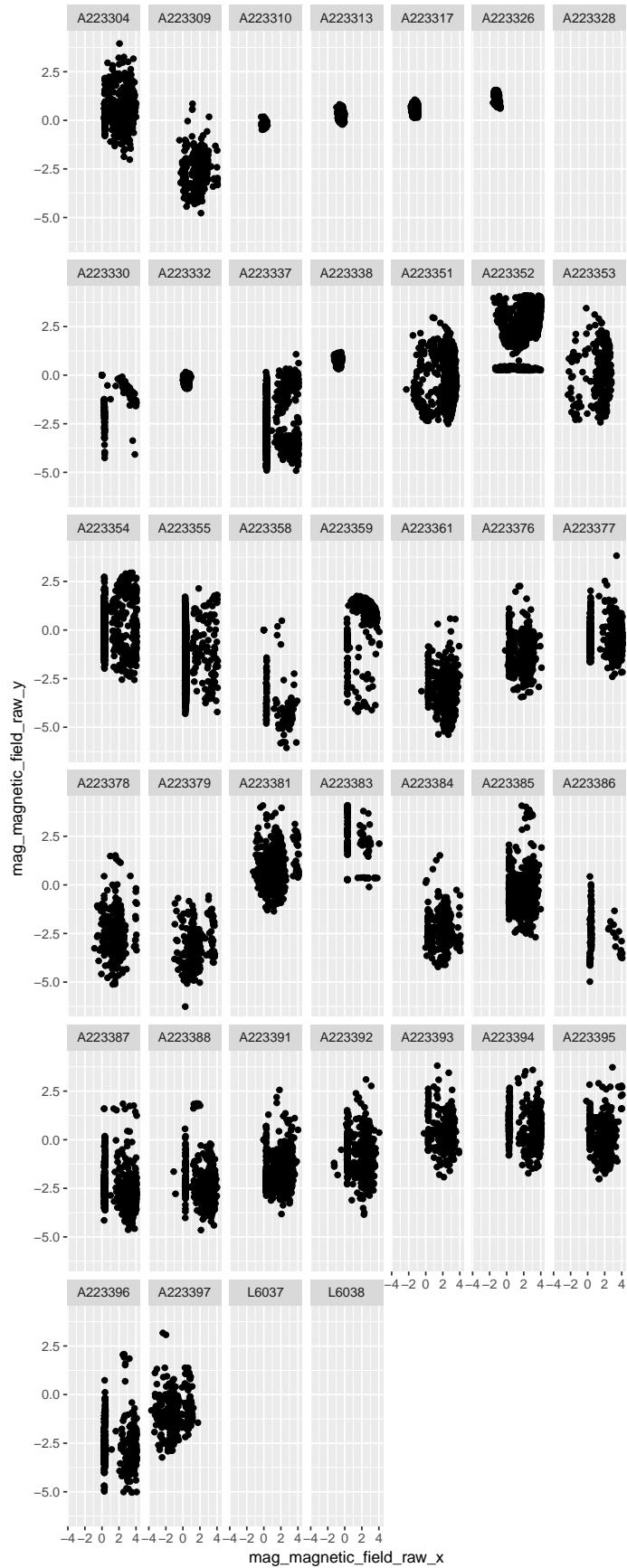
```
world_map_polygon <- map_data("world2")  
  
ggplot(data_crane) +  
  geom_map(data = world_map_polygon, map= world_map_polygon, aes(long, lat, map_id = region)) +  
  scale_x_continuous(limits = c(0, 60)) +  
  scale_y_continuous(limits = c(25, 60)) +  
  geom_point(data = data_crane, aes(x = location_long, y = location_lat, colour=individual_local_identi  
  
## Warning: Ignoring unknown aesthetics: x, y  
## Warning: Removed 200 rows containing missing values (geom_point).
```



Optional exercise (++) - Create facets.

```
ggplot(data_crane, aes(mag_magnetic_field_raw_x, mag_magnetic_field_raw_y)) +
  geom_point() +
  facet_wrap(~individual_local_identifier)

## Warning: Removed 432 rows containing missing values (geom_point).
```



3. Data transformation

Basic exercise I - Subset data (single crane)

a) Make a selection of all observations of crane ‘L6037’.

```
filter(data_crane, individual_local_identifier=='L6037')

## # A tibble: 61 x 42
##   event_id visible timestamp           location_long location_lat
##   <dbl>    <chr>   <dttm>              <dbl>        <dbl>
## 1 3.79e9  true    2017-10-15 00:00:16     40.3       55.3
## 2 3.80e9  true    2017-10-15 00:00:16     40.3       55.3
## 3 3.79e9  true    2017-10-15 02:00:23     40.3       55.3
## 4 3.80e9  true    2017-10-15 02:00:23     40.3       55.3
## 5 3.79e9  true    2017-10-15 04:00:23     40.3       55.3
## 6 3.80e9  true    2017-10-15 04:00:23     40.3       55.3
## 7 3.79e9  true    2017-10-15 05:00:14     40.1       55.3
## 8 3.80e9  true    2017-10-15 05:00:14     40.1       55.3
## 9 3.80e9  true    2017-10-15 06:00:23     40.1       55.3
## 10 3.80e9 true    2017-10-15 08:00:18      40.1       55.3
## # ... with 51 more rows, and 37 more variables: acceleration_raw_x <dbl>,
## #   acceleration_raw_y <dbl>, acceleration_raw_z <dbl>,
## #   bar_barometric_height <dbl>, battery_charge_percent <int>,
## #   battery_charging_current <dbl>, eobs_activity <chr>,
## #   eobs_activity_samples <chr>, eobs_battery_voltage <int>,
## #   eobs_fix_battery_voltage <int>,
## #   eobs_horizontal_accuracy_estimate <dbl>, eobs_key_bin_checksum <dbl>,
## #   eobs_speed_accuracy_estimate <dbl>, eobs_start_timestamp <chr>,
## #   eobs_status <chr>, eobs_temperature <int>, eobs_type_of_fix <int>,
## #   eobs_used_time_to_get_fix <int>, external_temperature <dbl>,
## #   gps_hdop <dbl>, gps_satellite_count <int>, gps_time_to_fix <dbl>,
## #   ground_speed <dbl>, heading <dbl>, height_above_ellipsoid <dbl>,
## #   height_above_msl <dbl>, import_marked_outlier <chr>,
## #   gls_light_level <dbl>, mag_magnetic_field_raw_x <dbl>,
## #   mag_magnetic_field_raw_y <dbl>, mag_magnetic_field_raw_z <dbl>,
## #   orn_transmission_protocol <chr>, tag_voltage <dbl>, sensor_type <chr>,
## #   individual_taxon_canonical_name <chr>,
## #   individual_local_identifier <chr>, study_name <chr>
```

b) Make a selection of all observations of crane ‘L6037’ where the variable eobs_status isn’t missing.

```
filter(data_crane, individual_local_identifier=='L6037', !is.na(eobs_status))

## # A tibble: 42 x 42
##   event_id visible timestamp           location_long location_lat
##   <dbl>    <chr>   <dttm>              <dbl>        <dbl>
## 1 3.80e9  true    2017-10-15 00:00:16     40.3       55.3
## 2 3.80e9  true    2017-10-15 02:00:23     40.3       55.3
## 3 3.80e9  true    2017-10-15 04:00:23     40.3       55.3
## 4 3.80e9  true    2017-10-15 05:00:14     40.1       55.3
## 5 3.80e9  true    2017-10-15 06:00:23     40.1       55.3
## 6 3.80e9 true    2017-10-15 08:00:18      40.1       55.3
```

```

## 7 3.80e9 true 2017-10-15 10:00:23 40.1 55.3
## 8 3.80e9 true 2017-10-15 12:00:14 40.1 55.3
## 9 3.80e9 true 2017-10-15 14:00:23 40.3 55.3
## 10 3.80e9 true 2017-10-15 16:00:23 40.3 55.3
## # ... with 32 more rows, and 37 more variables: acceleration_raw_x <dbl>,
## # acceleration_raw_y <dbl>, acceleration_raw_z <dbl>,
## # bar_barometric_height <dbl>, battery_charge_percent <int>,
## # battery_charging_current <dbl>, eobs_activity <chr>,
## # eobs_activity_samples <chr>, eobs_battery_voltage <int>,
## # eobs_fix_battery_voltage <int>,
## # eobs_horizontal_accuracy_estimate <dbl>, eobs_key_bin_checksum <dbl>,
## # eobs_speed_accuracy_estimate <dbl>, eobs_start_timestamp <chr>,
## # eobs_status <chr>, eobs_temperature <int>, eobs_type_of_fix <int>,
## # eobs_used_time_to_get_fix <int>, external_temperature <dbl>,
## # gps_hdop <dbl>, gps_satellite_count <int>, gps_time_to_fix <dbl>,
## # ground_speed <dbl>, heading <dbl>, height_above_ellipsoid <dbl>,
## # height_above_msl <dbl>, import_marked_outlier <chr>,
## # gls_light_level <dbl>, mag_magnetic_field_raw_x <dbl>,
## # mag_magnetic_field_raw_y <dbl>, mag_magnetic_field_raw_z <dbl>,
## # orn_transmission_protocol <chr>, tag_voltage <dbl>, sensor_type <chr>,
## # individual_taxon_canonical_name <chr>,
## # individual_local_identifier <chr>, study_name <chr>

```

c) Make a selection of all observations of crane ‘L6037’ where the variable `eobs_status` isn’t missing. Return only the `event_id` and `timestamp` variables.

```

data_crane_filter <- filter(data_crane, individual_local_identifier=='L6037')
select(data_crane_filter, event_id, timestamp)

```

```

## # A tibble: 61 x 2
##       event_id timestamp
##   <dbl> <dttm>
## 1 3794510958 2017-10-15 00:00:16
## 2 3796034354 2017-10-15 00:00:16
## 3 3794510962 2017-10-15 02:00:23
## 4 3796034355 2017-10-15 02:00:23
## 5 3794510961 2017-10-15 04:00:23
## 6 3796034356 2017-10-15 04:00:23
## 7 3794510960 2017-10-15 05:00:14
## 8 3796034357 2017-10-15 05:00:14
## 9 3796034358 2017-10-15 06:00:23
## 10 3796034359 2017-10-15 08:00:18
## # ... with 51 more rows

```

Basic exercise II - Compute the magnitude of the magnetic field

```

data_crane_with_mag_magnetic <- mutate(
  data_crane,
  magnitude = sqrt(mag_magnetic_field_raw_x ^ 2 +
    mag_magnetic_field_raw_y ^ 2 +
    mag_magnetic_field_raw_z ^ 2)
)

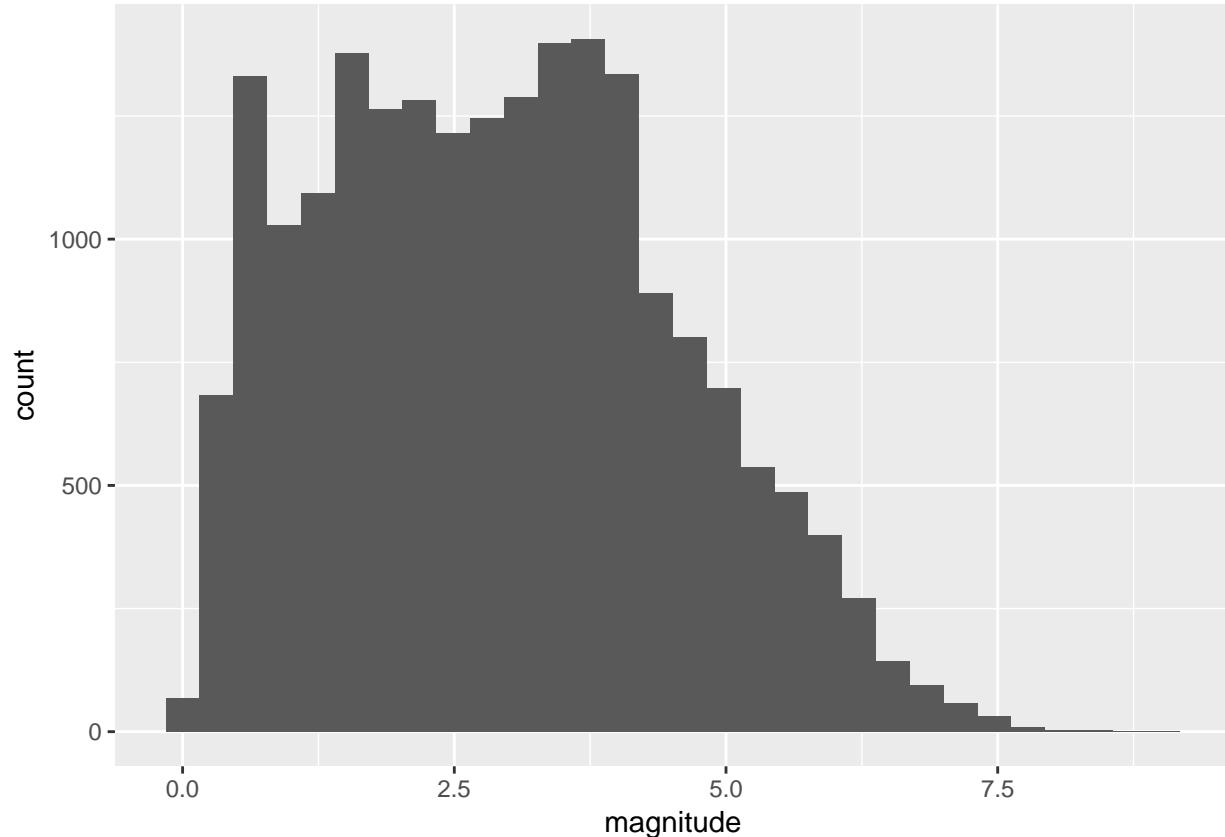
```

```

ggplot(data_crane_with_mag_magnetic) +
  geom_histogram(aes(magnitude))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 432 rows containing non-finite values (stat_bin).

```



Reading exercise - Pipe operator

The pipe operator `%>%` is used by many tidyverse users. Read Chapter 18 of R for Data Science. Make a selection columns and rows of the data and make use of the pipe syntax.

Optional exercise (+) - Exclude variables

```

select(data_crane, -starts_with('location'))

## # A tibble: 20,873 x 40
##   event_id visible timestamp      acceleration_ra~ acceleration_ra~
##   <dbl>    <chr>   <dttm>          <dbl>          <dbl>
## 1 3.79e9  true   2017-10-15 00:00:16     NA            NA
## 2 3.80e9  true   2017-10-15 00:00:16     NA            NA
## 3 3.79e9  true   2017-10-15 02:00:23     NA            NA
## 4 3.80e9  true   2017-10-15 02:00:23     NA            NA
## 5 3.79e9  true   2017-10-15 04:00:23     NA            NA
## 6 3.80e9  true   2017-10-15 04:00:23     NA            NA

```

```

## 7 3.79e9 true 2017-10-15 05:00:14 NA NA
## 8 3.80e9 true 2017-10-15 05:00:14 NA NA
## 9 3.80e9 true 2017-10-15 06:00:23 NA NA
## 10 3.80e9 true 2017-10-15 08:00:18 NA NA
## # ... with 20,863 more rows, and 35 more variables:
## #   acceleration_raw_z <dbl>, bar_barometric_height <dbl>,
## #   battery_charge_percent <int>, battery_charging_current <dbl>,
## #   eobs_activity <chr>, eobs_activity_samples <chr>,
## #   eobs_battery_voltage <int>, eobs_fix_battery_voltage <int>,
## #   eobs_horizontal_accuracy_estimate <dbl>, eobs_key_bin_checksum <dbl>,
## #   eobs_speed_accuracy_estimate <dbl>, eobs_start_timestamp <chr>,
## #   eobs_status <chr>, eobs_temperature <int>, eobs_type_of_fix <int>,
## #   eobs_used_time_to_get_fix <int>, external_temperature <dbl>,
## #   gps_hdop <dbl>, gps_satellite_count <int>, gps_time_to_fix <dbl>,
## #   ground_speed <dbl>, heading <dbl>, height_above_ellipsoid <dbl>,
## #   height_above_msl <dbl>, import_marked_outlier <chr>,
## #   gls_light_level <dbl>, mag_magnetic_field_raw_x <dbl>,
## #   mag_magnetic_field_raw_y <dbl>, mag_magnetic_field_raw_z <dbl>,
## #   orn_transmission_protocol <chr>, tag_voltage <dbl>, sensor_type <chr>,
## #   individual_taxon_canonical_name <chr>,
## #   individual_local_identifier <chr>, study_name <chr>

```

Optional exercise (++) - Summarise results

```

summarise(
  data_crane,
  min_latitude = max(location_lat, na.rm=T),
  first_observation = min(timestamp),
  magnitude_acceleration = mean(sqrt(acceleration_raw_x^2 + acceleration_raw_y^2 + acceleration_raw_z^2))
)

## # A tibble: 1 x 3
##   min_latitude first_observation   magnitude_acceleration
##       <dbl>          <dttm>                      <dbl>
## 1      55.4 2017-10-15 00:00:05           516.

```

Optional exercise (+++) - Join datasets

```
data_crane_with_measures <- left_join(data_crane, data_crane_additional, by='event_id')
```