# Software for transport data science

Robin Lovelace University of Leeds, 2020-02-04

UNIVERSITY OF LEEDS
Institute for Transport Studies (ITS)

# Project set-up and tidyverse testing (30 minutes, individually)

- Check your packages are up-to-date with `update.packages()`
- Create an RStudio project with an appropriate name for this module (e.g. `TDS` )
- Create appropriate files for code, data and anything else (e.g. images)
- Create a script called `learning-tidyverse.R` , e.g. with **one** the following commands:

```
file.edit(learning-tidyverse.R) # or
file.edit(code/learning-tidyverse.R)
```

- Read section 5.1 (https://r4ds.had.co.nz/transform.html#filter-rows-with-filter) of R for Data Science and write code that reproduces the results in that section in the script `learning-tidyverse.R`

Your script will start with something like this:

```
library(tidyverse)
```

```
## ── Attaching packages ───────────────────────────────────────────
─────────────────────────────────────── tidyverse 1.3.0 ──

## ✔ ggplot2 3.2.1     ✔ purrr   0.3.3
## ✔ tibble  2.1.3     ✔ dplyr   0.8.3
## ✔ tidyr   1.0.2     ✔ stringr 1.4.0
## ✔ readr   1.3.1     ✔ forcats 0.4.0

## ── Conflicts ─────────────────────────────────────────────────────
───────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(nycflights13)
```

# Reading-in and processing coffee data

Read-in the coffee data we created last week, e.g. with:

```
u = paste0(
  "https://github.com/ITSLeeds/TDS/",
  "raw/master/sample-data/everyone.csv"
  )
d = read_csv(u)
```

```
## Parsed with column specification:
## cols(
##   person_name = col_character(),
##   n_coffee = col_double(),
##   like_bus_travel = col_logical()
## )
```

Create a new variable called ‘n_coffee_yr’ with the following command:

```
d$n_coffee_yr = d$n_coffee * 52
```

Find the mean number of cups of coffee people drink per year (and the total)

Note: the same result can be acheived as follows:

```
d = mutate(d, n_coffee_yr = n_coffee * 52)

# or
d = d %>%
  mutate(n_coffee_yr = n_coffee * 52)
```

- Which do you prefer?
- Filter-out only those who travel by bus
- Bonus: Create a new dataset that keeps only the `person_name` and `n_coffee_yr` variables (hint: use the `select()` function)
- Bonus: do those who travel by bus drink more or less coffee than those who do not?

# Processing a big file and basic visualisation (30 minutes, individually)

- Take a random sample of 10,000 flights and assign it to an object with the following line of code:

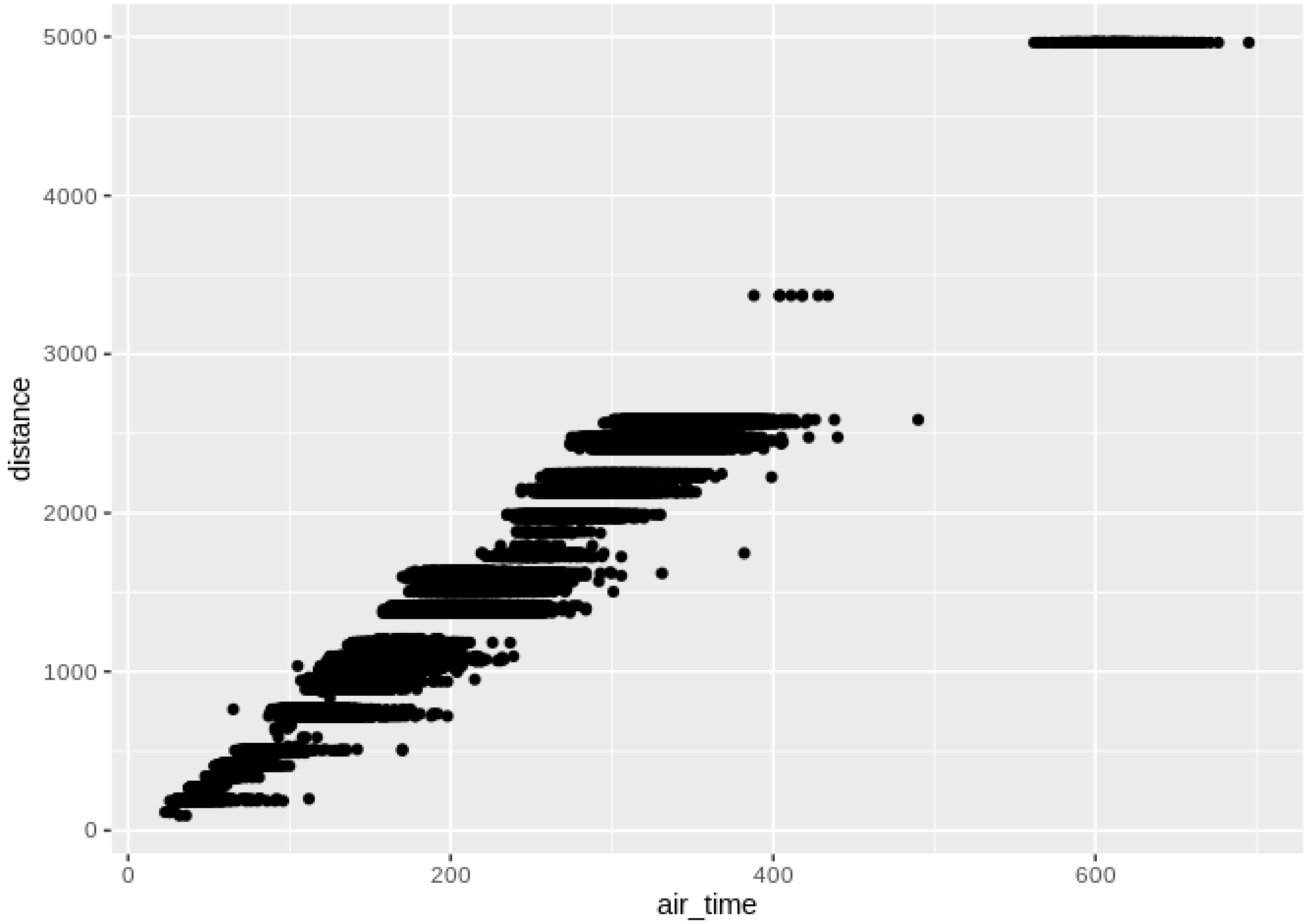
```
flights_sample = sample_n(flights, 1e4)
```

- Find the unique carriers with the `unique()` function
- Create an object containing flights from United, American, or Delta, and assign it to `f` , as follows:

```
f = filter(flights, grepl(pattern = "UA|AA|DL", x = carrier))
```

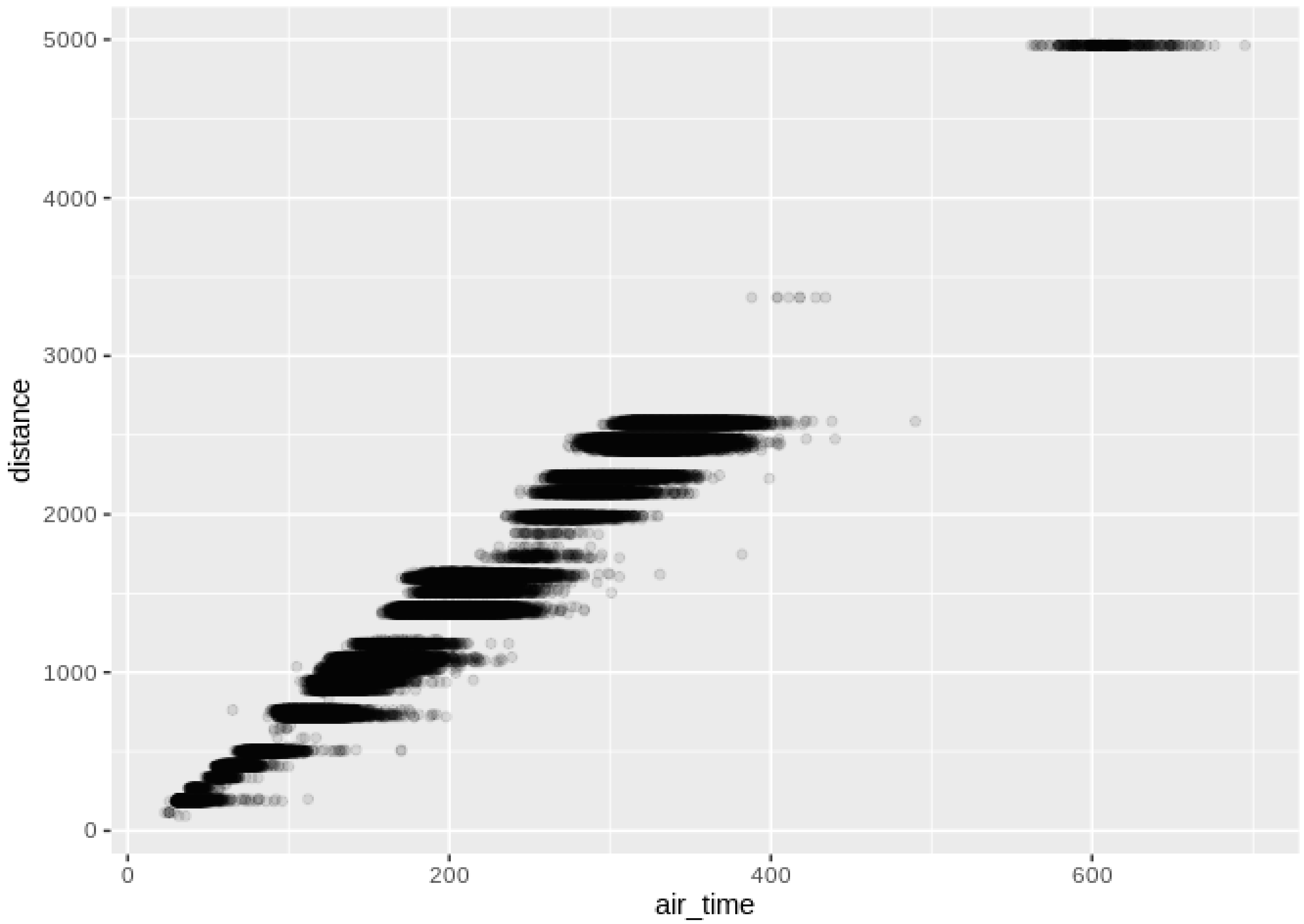
- Create plots that visualise the sample flights, using code from Chapter 3 of the same book, starting with the following plot:

```
ggplot(f) +
  geom_point(aes(air_time, distance))
```



- Add transparency so it looks like this (hint: use `alpha =` in the `geom_point()` function call):

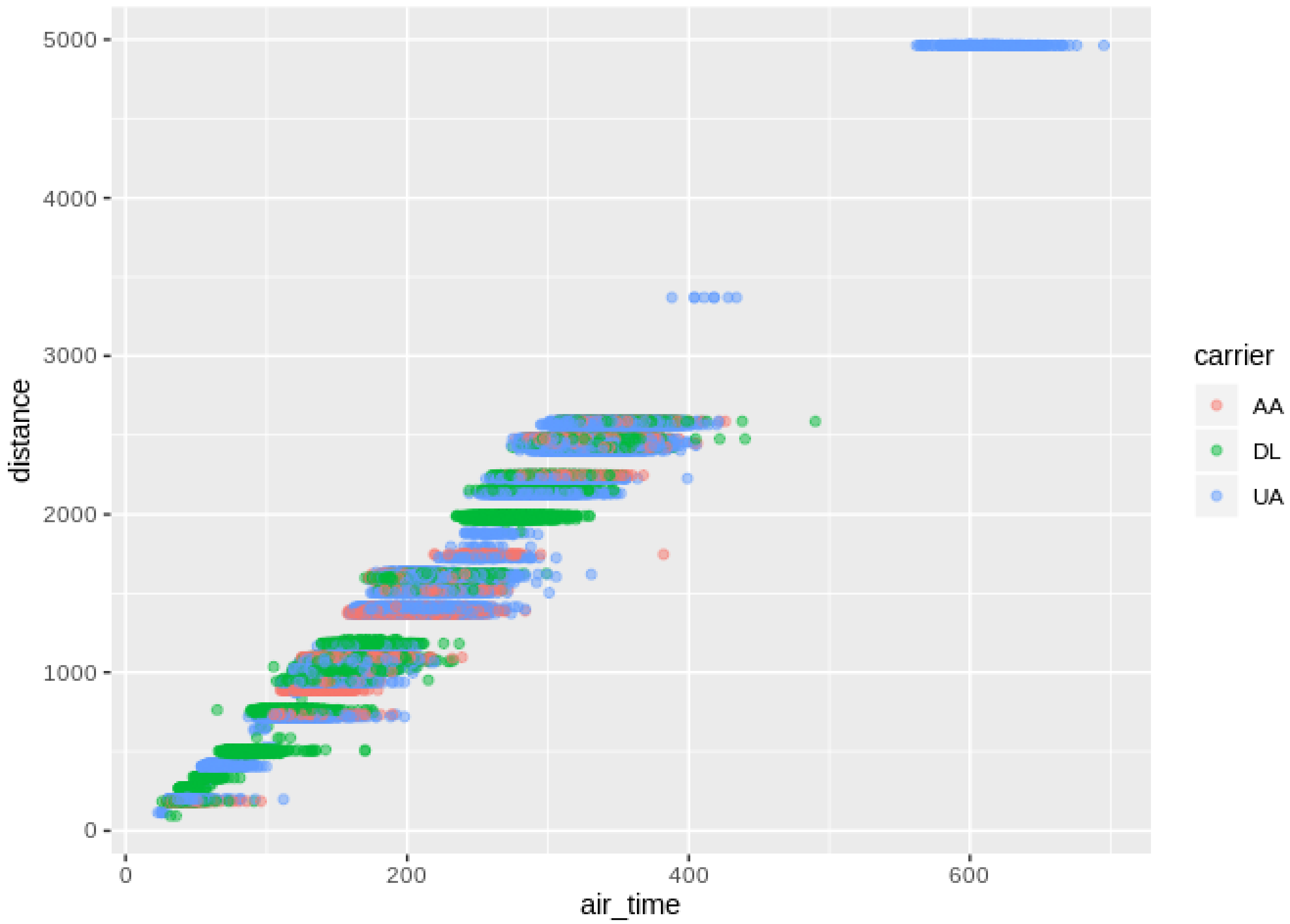
```
## Warning: Removed 2117 rows containing missing values (geom_point).
```



- Add a colour for each carrier, so it looks something like this:

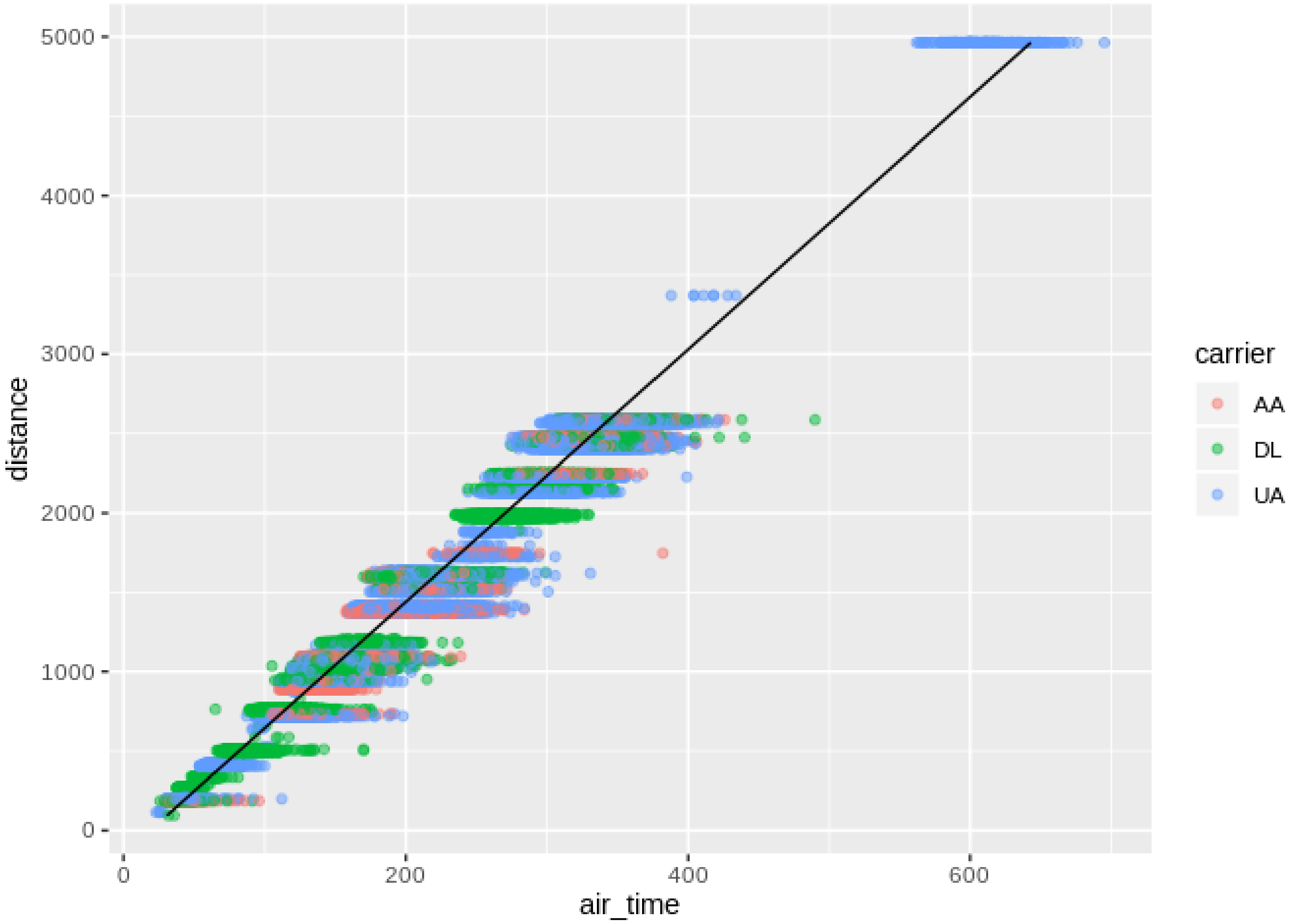
```
ggplot(f) +
  geom_point(aes(air_time, distance, colour = carrier), alpha = 0.5)
```

```
## Warning: Removed 2117 rows containing missing values (geom_point).
```



- Bonus 1: find the average air time of those flights with a distance of 1000 to 2000 miles
- Bonus 2: use the `lm()` function to find the relationship between flight distance and time, and plot the results (start the plot as follows, why did we use `na.omit()` ? hint - find help with `?na.omit()` ):

```
f = na.omit(f)
m = lm(air_time ~ distance, data = f)
f$pred = m$fitted.values
```



# Homework

1. create a reproducible document

- Create an Rmarkdown file with the following command:

```
file.edit("learning-tidyverse.Rmd")
```

- Take a read of the guidance on RMarkdown files online and in the following location (or search online for the 'RMarkdown cheatsheet'):

```
Help > Cheatsheets > RMarkdown
```

- Put the code you generated for `tidyverse.R` into the Rmd file and knit it
- Bonus: create a GitHub repo and publish the results of of your work (hint: putting `output: github_document` may help here!)

2. Work-through the remaining exercises of the first sections in R4DS chapters 3 and 5

- Write and R script, with comments, to show your working (and prove you've done it!)

3. Create an RMarkdown file containing reproducible code outlining what you learned today
4. Identify a dataset you would like to work with for the practical next week.