

WEEK 3

RELATIONSHIPS & COMPARISONS

DATA VISUALIZATION FOR SOCIAL SCIENTISTS

LECTURER: JEFFREY ZIEGLER, PhD

TEACHING FELLOW: SHEKHAR KEDIA

ASDS - TRINITY COLLEGE DUBLIN

SPRING 2026

ROAD MAP FOR TODAY

- Today:
 - ▶ Dual y-axes in `ggplot()`
 - ▶ Combining plots with `patchwork`
 - ▶ Correlations, scatterplot matrices, and correlograms
 - ▶ Regression, predictions, and marginal effectsd plot them with `ggplot2`
- By next week, please...
 - ▶ Problem set #3

LEGAL DUAL Y-AXES

- Safe when both axes measure same underlying quantity
 - ▶ Ex: counts vs %, Fahrenheit vs Celsius, pounds vs kilograms
- In ggplot2: use `sec.axis` inside `scale_y_continuous()`

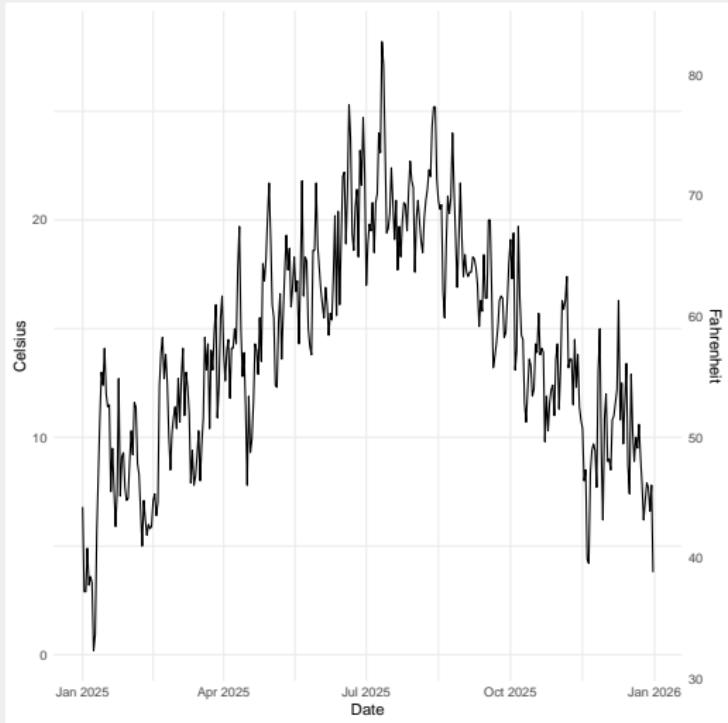
Ex: DUAL Y-AXES: FAHRENHEIT & CELSIUS

- Historical weather for Dublin: Met Éireann
- Variables include temperature, humidity, wind speed, precipitation, etc
space*.25cm
- C to F conversion: $F = (C) \times \frac{9}{5} + 32$

Ex: MAX TEMP BY DAY IN 2025 FOR DUBLIN

```
1 # we'll only look at 2025
2 DUB_weather <- DUB_weather[30317:
  nrow(DUB_weather),]
3 CORK_weather <- CORK_weather[23012:
  nrow(CORK_weather),]
4 IRE_weather <- rbind(subset(DUB_
  weather, select=-c(g_rad)),
  CORK_weather)
5 # need to alter date variable to
  not be character
6 IRE_weather$date <- dmy(DUB_weather
  $date)

1 ggplot(IRE_weather[IRE_weather$`station`=="Dublin",], aes(x =
  date, y = maxtp)) +
2   geom_line() +
3   scale_y_continuous(sec.axis = sec
  _axis(trans = ~ (. * 9/5) +
  32, name = "Fahrenheit")) +
4   labs(x = "Date", y = "Celsius") +
5   theme_minimal()
```

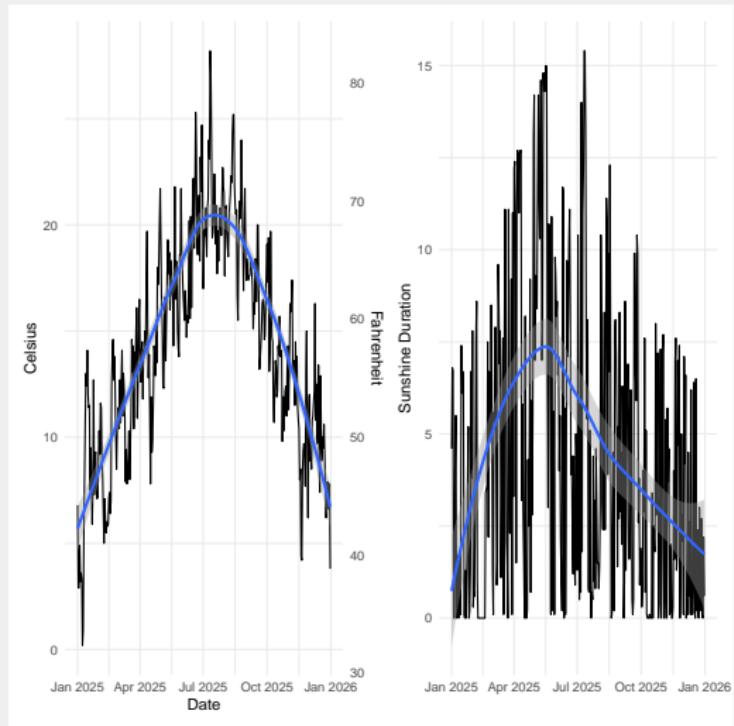


COMBINING PLOTS WITH PATCHWORK

- Alternative to dual axes: separate, aligned plots
- patchwork combines ggplots
 - ▶ To use patchwork, we need to (1) save our plots as objects and (2) add them together with +
- Supports layouts, relative heights, and more

Ex: RELATIONSHIP OF TEMP & SUNSHINE DURATION

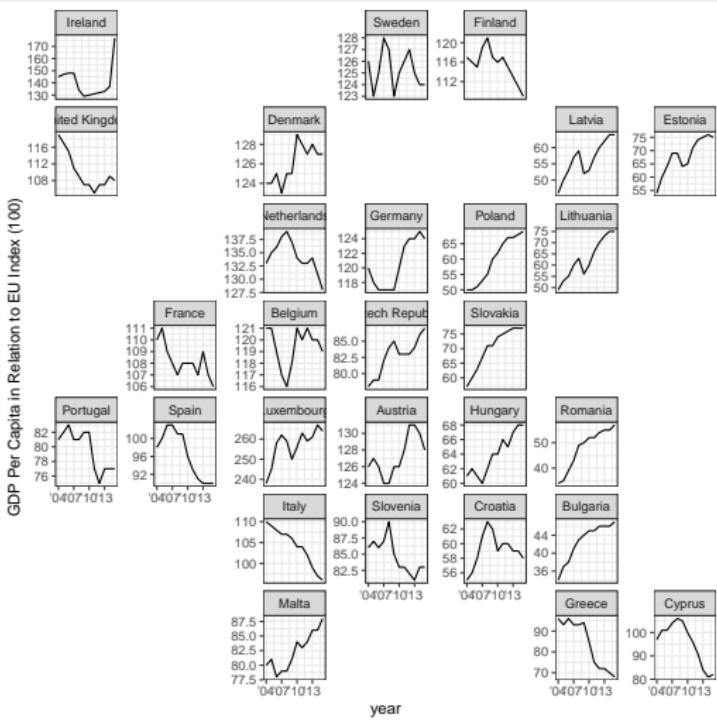
```
1 sun_plot <- ggplot(IRE_weather[IRE_weather$station=="Dublin",],  
2   aes(x = date, y = sun)) +  
3   geom_line() +  
4   geom_smooth() +  
5   labs(x = NULL, y = "Sunshine Duration") +  
6   theme_minimal()  
  
1 # library(patchwork)  
2 temp_plot + sun_plot
```



COMPARING TIME SERIES

- Many-group (i.e., country, city) line charts can become difficult to read
- Strategies:
 - ▶ Highlight a few focal groups
 - ▶ Use small multiples (facets)
 - ▶ Use map-based facet layouts with geofacet
 - geofacet arranges country facets in map-like grids
 - Preserves some geographic structure while showing small multiples
 - Useful for regional or global comparisons

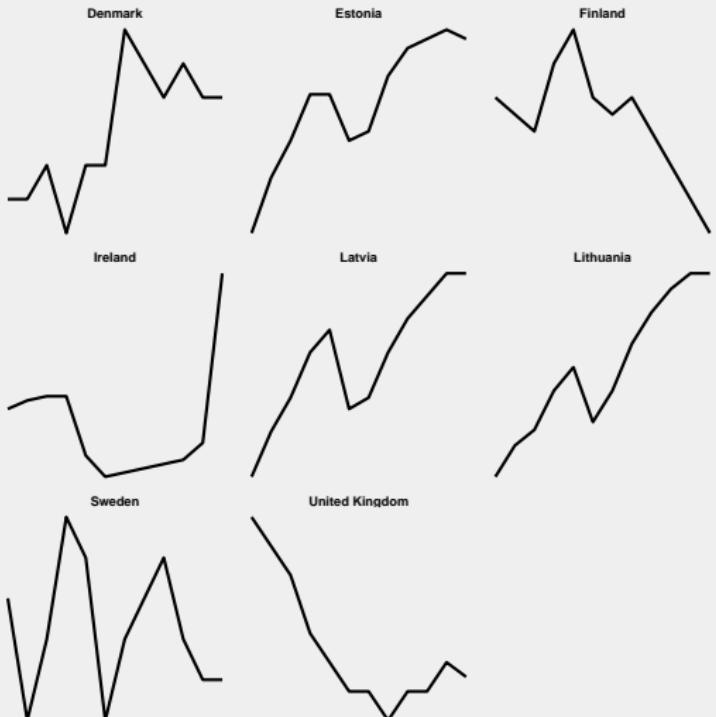
GEOFACET EXAMPLE



```
1 # library("geofacet")
2 ggplot(eu_gdp, aes(year, gdp_pc)) +
3   geom_line() +
4   facet_geo(~ name, grid = "eu_grid1", scales = "free_y") +
5   scale_x_continuous(labels =
6     function(x) paste0("", substr
(x, 3, 4))) +
7   ylab("GDP Per Capita in Relation
to EU Index (100)") +
7   theme_bw()
```

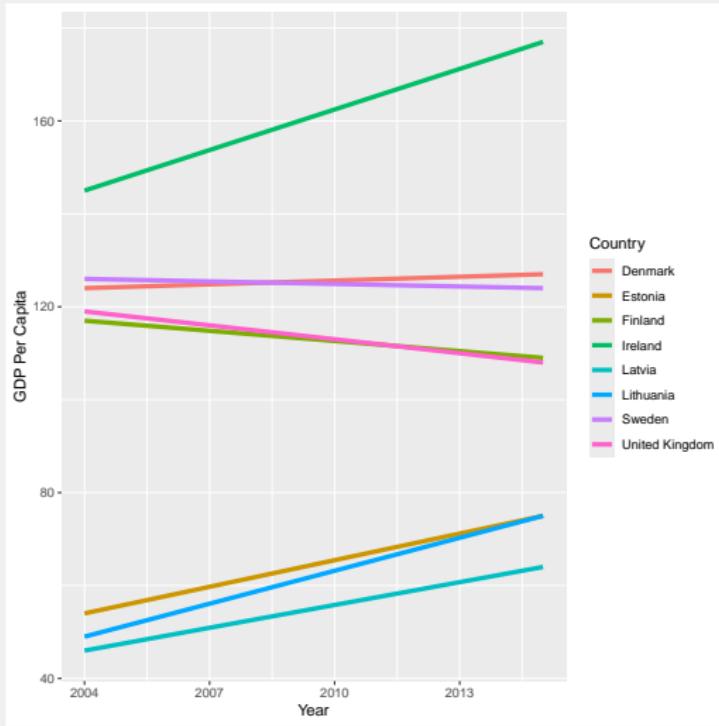
Ex: SMALL MULTIPLES - EU GDP IN NORTHERN EUROPE

```
1 # create "Northern Europe"
  indicator based on UN
  geoscheme
2 eu_gdp$north_europe <- ifelse(eu_
  gdp$name %in% c("Denmark", "Estonia", "Finland", "Ireland", "Latvia", "Lithuania", "Norway", "Sweden", "United Kingdom"), 1, 0)
3
4 ggplot(data = eu_gdp |> filter(
  north_europe == 1), aes(x =
  year, y = gdp_pc)) +
  geom_line(linewidth = 1) +
  facet_wrap(vars(name), scales =
    "free_y", nrow = 3) +
  theme_void() +
  theme(strip.text = element_text(
    face = "bold"))
```



SLOPEGRAPHS

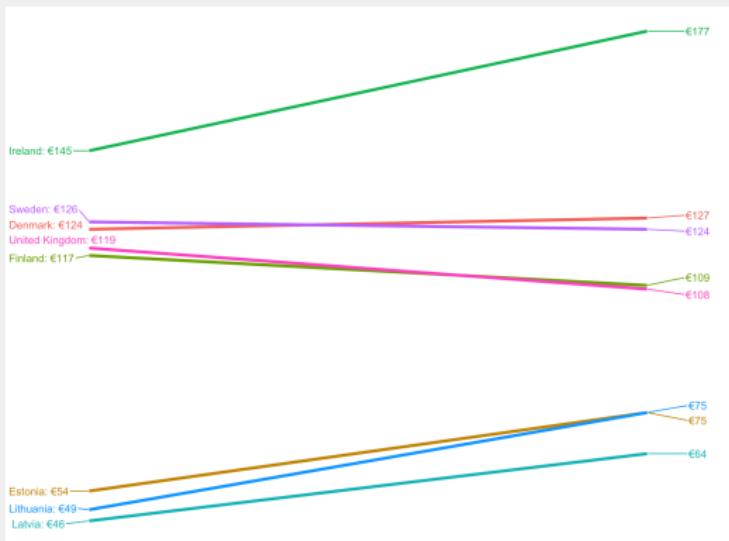
Show only change in GDP per capita between two time periods



```
1 north_europe_gdp <- eu_gdp |>
  filter(north_europe == 1) |>
  filter(year %in% c(2004, 2015)) |>
  mutate(label_first = ifelse(year == 2004, paste0(name, ":", label_dollar(prefix = " ")(round(gdp_pc))), NA),
        label_last = ifelse(year == 2015, label_dollar(prefix = " ")(round(gdp_pc, 0)), NA))
  )
4 ggplot(north_europe_gdp, aes(x = year, y = gdp_pc, group = name,
  , color = name)) +
5   geom_line(linewidth = 1.5) +
6   labs(y = "GDP Per Capita", x = "Year",
  , color = "Country")
```

LABELING SELECTED POINTS WITH GGREPEL

```
1 ggplot(north_europe_gdp, aes(x =
2   year, y = gdp_pc, group = name))
3   , color = name)) +
4     geom_line(linewidth = 1.5) +
5     geom_text_repel(aes(label = label
6       _first), direction = "y",
7         nudge_x = -1, seed = 1234) +
8     geom_text_repel(aes(label = label
9       _last), direction = "y", nudge
10      _x = 1, seed = 1234) +
11     guides(color = "none") +
12     theme_void()
```



DESIGN CONSIDERATIONS

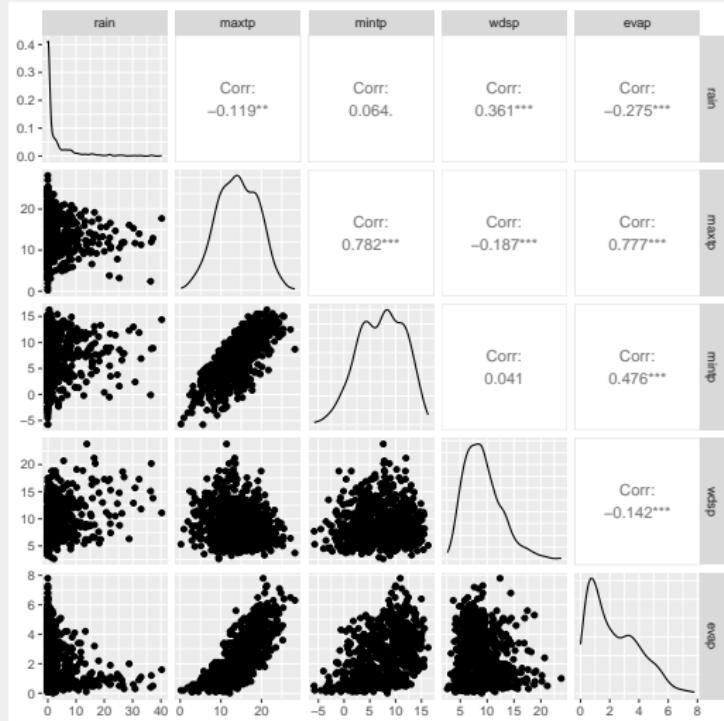
- Use sorting and consistent scales to clarify comparisons
- Prefer small multiples over overloaded single panels
- Highlight only a few key labels; avoid label clutter
- Use log scales when data span orders of magnitude

EXPLORING CORRELATIONS

- Use `GGally::ggpairs()` for scatterplot matrices
- Visualize relationships among several variables at once
- Good for exploration; often too dense for publication

SCATTERPLOT MATRIX WITH GGPALS

```
1 # library(GGally)
2 weather_cor <- IRE_weather |> select(rain, maxtp, mintp, wdsp, evap)
```



CORRELATION PATTERNS

- High vs low temperature: very strong positive correlation ($r \approx 0.78$)
- High temp vs evaporation: very strong positive correlation ($r \approx 0.77$)
- Wind speed vs rain: moderate positive correlation
- Wind speed vs temperature: minor negative correlation
- Little or no correlation for some variable pairs (e.g., wind speed vs min temp)

CORRELOGRAMS

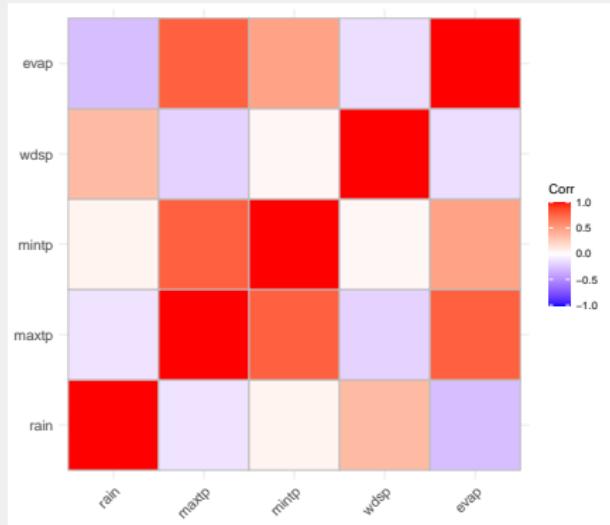
- Heatmaps of correlation coefficients
- More suitable for publication than full scatterplot matrices
- Depending on package, need to compute a correlation matrix and tidy it

Ex: CORRELOGRAM WITH WEATHER

```
1 # library(corrgram)  
2 corrgram(weather_cor)
```



```
1 # library(ggcorrplot)  
2 ggcorrplot(cor(weather_cor))
```



SIMPLE REGRESSION: IDEA

- Outcome: Daily high temperature
- Predictor: Rain

```
1 # run our "simple" regression w/ max temp as outcome
2 model_simple <- lm(maxtp ~ rain, data = IRE_weather)
3 # library(broom)
4 tidy(model_simple, conf.int = TRUE)
```

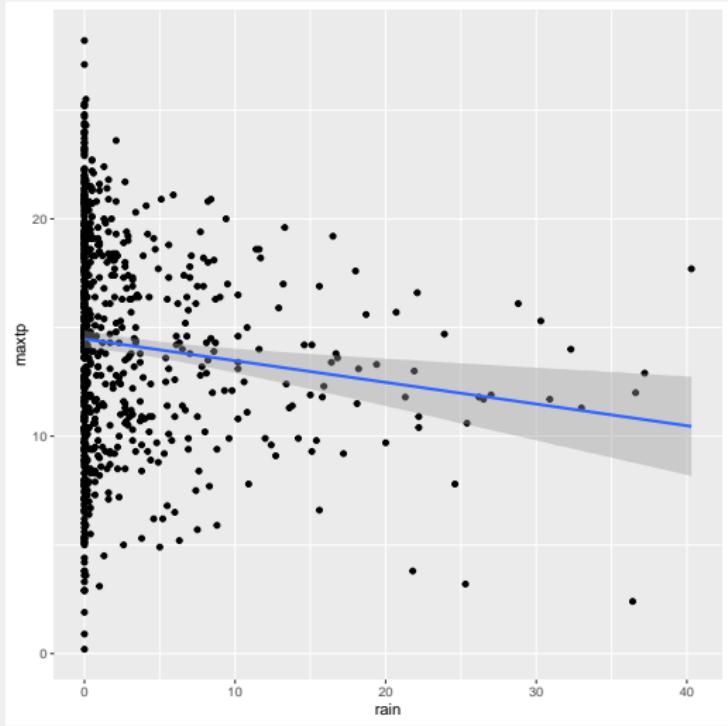
```
# A tibble: 2 × 7
term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>     <dbl>      <dbl>    <dbl>     <dbl>     <dbl>
1 (Intercept) 14.5      0.204     70.9     0       14.1     14.9
2 rain        -0.0997   0.0307    -3.24    0.00124  -0.160    -0.0393
```

We can interpret these coefficients like so:

- Intercept: Average temperature when there's no rain is 14.5°C
- β_{rain} : ↑ 1mm of rain is associated with a 0.0997° decrease in max temp, on average

VISUALIZING SIMPLE REGRESSION

```
1 ggplot(IRE_weather,  
2         aes(x = rain, y = maxtp)) +  
3         geom_point() +  
4         geom_smooth(method = "lm")
```



MULTIPLE REGRESSION

- Add predictors: humidity, moon phase, rain probability, wind speed, pressure, cloud cover
- Hard to visualize directly in original space
- Use coefficient plots and predicted values

MULTIPLE REGRESSION

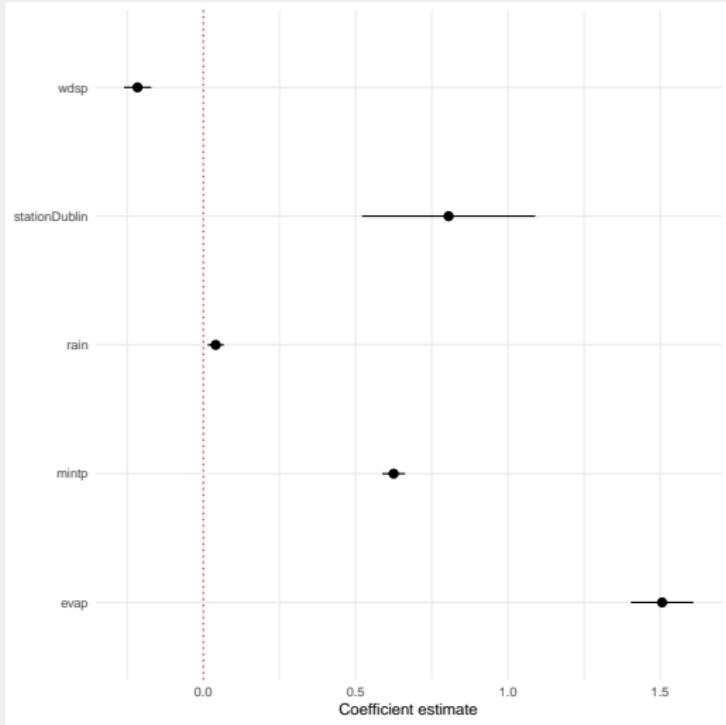
```
1 # run our "complex" regression w/ max temp as outcome
2 model_complex <- lm(maxtp ~ rain + mintp + wdsp + evap + station ,
3                      data = IRE_weather)
4 tidy(model_complex, conf.int = TRUE)

# A tibble: 6 × 7
term      estimate std.error statistic p.value conf.low conf.high
<chr>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept) 7.70     0.267     28.8  3.38e-122   7.17     8.22
2 rain        0.0406    0.0136     2.98  3.00e- 3   0.0138   0.0674
3 mintp       0.625     0.0192     32.6  2.87e-144   0.587     0.663
4 wdsp        -0.216    0.0228    -9.46  4.16e- 20  -0.261    -0.171
5 evap         1.51     0.0521     28.9  8.80e-123   1.40      1.61
6 stationDublin 0.805    0.145      5.56  3.79e- 8   0.521     1.09
```

- β_{rain} : On average, $\uparrow 1\text{mm}$ of rain is associated with a 0.0406°C increase in max temp, holding all else constant
- β_{Dublin} : On average, Dublin has a 0.805°C higher max temp than Cork, holding all else constant

COEFFICIENT PLOT

```
1 ggplot(tidy(model_complex, conf.int
  = TRUE) |> filter(term != "(Intercept)"),
2         aes(x = estimate, y = term)) +
3   geom_vline(xintercept = 0, color
4             = "red", linetype = "dotted") +
5   geom_pointrange(aes(xmin = conf.
6     low, xmax = conf.high)) +
7   labs(x = "Coefficient estimate",
8        y = NULL) +
9   theme_minimal()
```

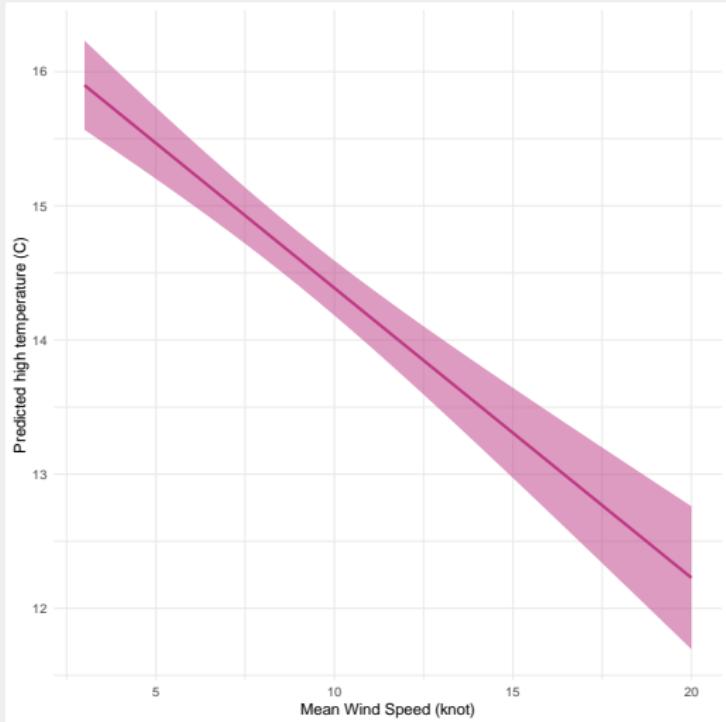


USING MARGINALEFFECTS

- `predictions()` for predicted values with CIs
- `slopes()` for marginal effects (slopes)
- `datagrid()` constructs scenarios, holding other variables at typical values

MARGINALEFFECTS - WIND

```
1 # library(marginaleffects)
2 # Calculate predictions across a
#      range of windSpeed
3 predicted_values_easy <-
  predictions(model_complex,
4   newdata = datagrid(wdsp = seq(3,
5     20, 0.5)))
5 )  
  
1 ggplot(predicted_values_easy, aes(x
#      = wdsp, y = estimate)) +
2   geom_ribbon(aes(ymin = conf.low,
#      ymax = conf.high),
3     fill = "#BF3984",
4     alpha = 0.5) +
5   geom_line(linewidth = 1, color =
#BF3984") +
6   labs(x = "Mean Wind Speed (knot)"
#      , y = "Predicted high
7     temperature (C)") +
```



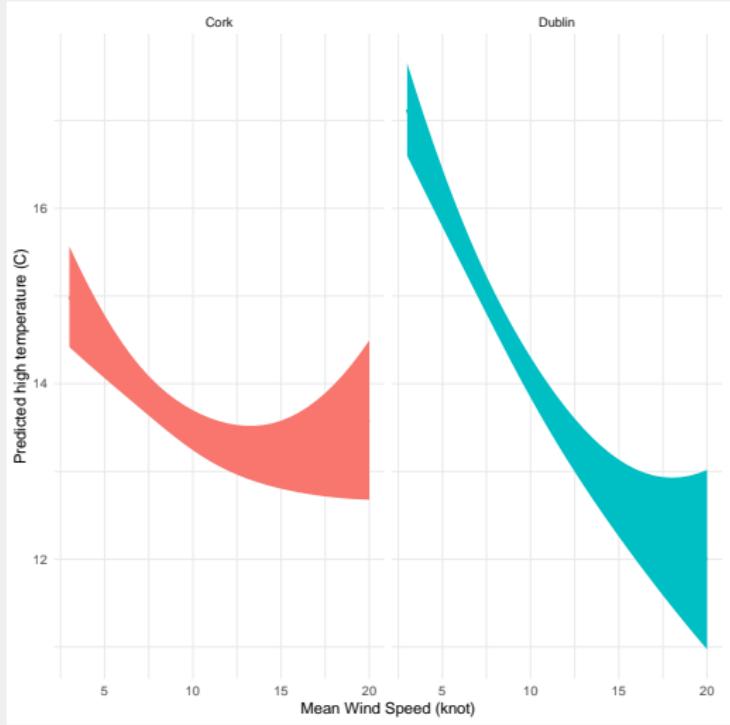
NONLINEAR AND INTERACTION MODEL

You could:

- Add quadratic term for wind speed
- Add interaction: wind speed \times station

```
1 model_wild <- lm(maxtp ~ rain + mintp + wdsp + evap + station + I(wdsp^2) + wdsp:station ,  
                     data = IRE_weather)  
2  
3 predicted_values_wild <- predictions(  
4   model_wild ,  
5   newdata =datagrid(  
6     wdsp = seq(3, 20, 0.5),  
7     station = c("Cork", "Dublin")))
```

MARGINALEFFECTS - INTERACTIONS



```
1 ggplot(predicted_values_wild, aes(x  
= wdsp, y = estimate)) +  
2   geom_ribbon(aes(ymin = conf.low,  
      ymax = conf.high, fill =  
      station)) +  
3   geom_line(aes(color = station),  
      linewidth = 1) +  
4   labs(x = "Mean Wind Speed (knot)"  
      , y = "Predicted high  
      temperature (C)") +  
5   theme_minimal() +  
6   guides(fill = "none", color = "  
      none") +  
7   facet_wrap(vars(station), nrow =  
      1)
```

WRAP UP

- Dual y-axes are safe when measuring the same quantity in different units
- patchwork simplifies multi-plot layouts
- Correlograms give compact views of many correlations
- broom + marginaleffects streamline predictions and marginal effects

CLASS BUSINESS

- Read required (and suggested) online materials
- Problem set #3 is up on GitHub