MONASH University

MONASH BUSINESS SCHOOL

Introduction to Data Analysis and Visualisation using R

Professor Di Cook, Econometrics and Business Statistics

Workshop for the Institute for Safety, Compensation and Recovery Research

GROUP OF EIGHT AUSTRALIA

AACSB ACCREDITED EFMD EQUIS ACCREDITED ASSOCIATION AMBA ACCREDITED

Session 4

Advanced graphics, layering, using maps

(If you re-started RStudio, be sure to re-open your project too.)

NYC Workers Compensation Claims

To examine the temporal trend of claims, the strudture of the data is:

- Basic unit is a claim case
- Multiple cases each day

To organise it:

- Aggregate to day level
- Plot count against day

Read the data

```
workers <- read_csv(
  file="data/Assembled_Workers__Compensation_Claims___Beginnir
workers$`Accident Date` <- as.Date(workers$`Accident Date`,
                                   format="%m/%d/%Y")
```

Extract temporal components

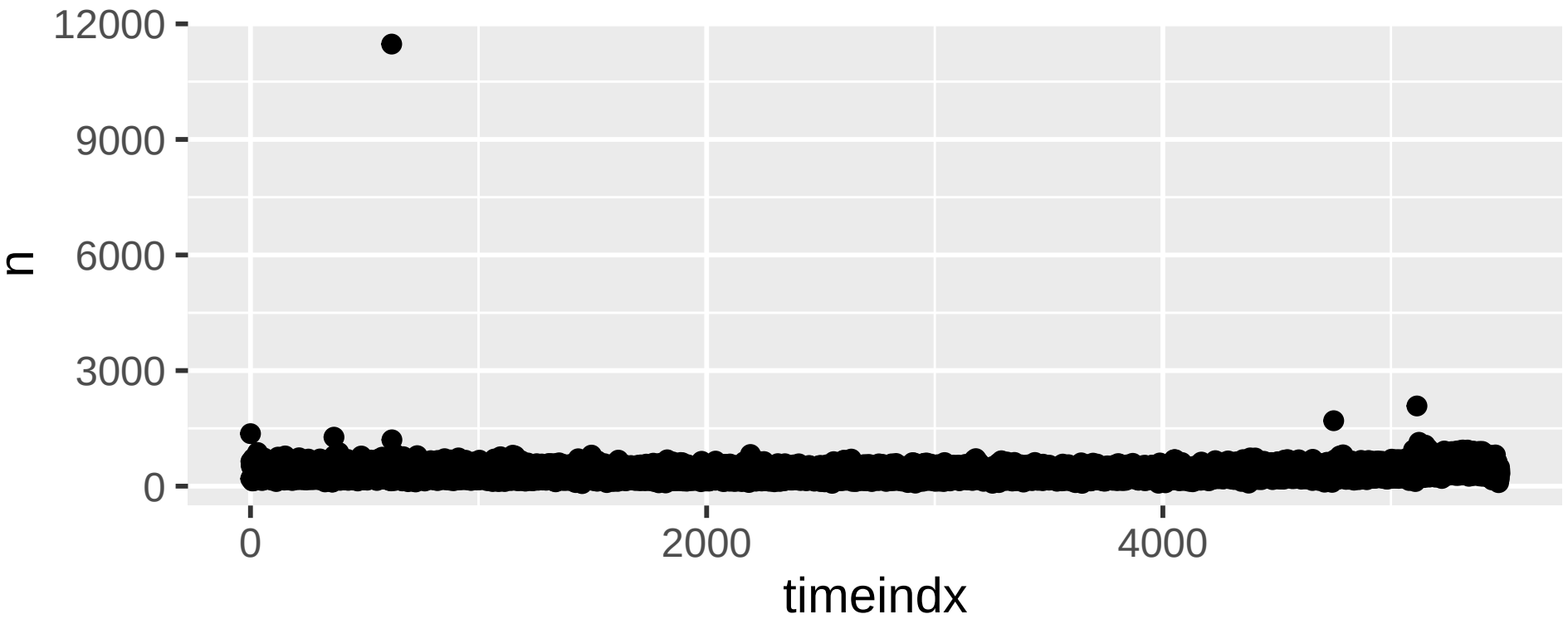
```
workers$year <- year(workers$`Accident Date`)
workers$month <- month(workers$`Accident Date`,
                       label=TRUE, abbr=TRUE)
workers$wday <- wday(workers$`Accident Date`,
                     label=TRUE, abbr=TRUE)
workers$timeindx <- as.numeric(workers$`Accident Date`-
        as.Date("01/01/2000", format="%m/%d/%Y"))
```

Filter, re-order, tally cases

```
ws <- workers %>% filter(year > 1999 & year < 2015)
ws$wday <- factor(ws$wday, levels=levels(ws$wday)[c(2:7,1)])
wsd <- ws %>% group_by(timeindx) %>% tally()
```

Plot it

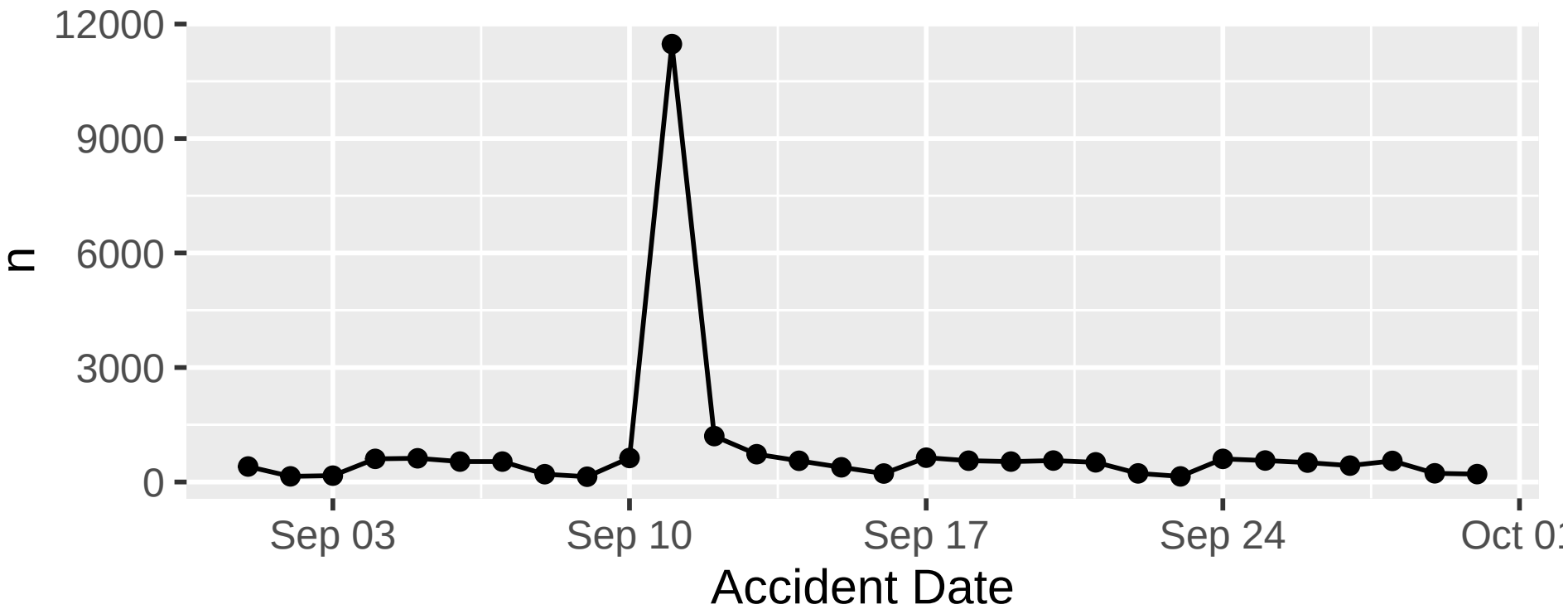
```
ggplot(wsd, aes(timeindx, n)) + geom_point()
```



!!! what is that?

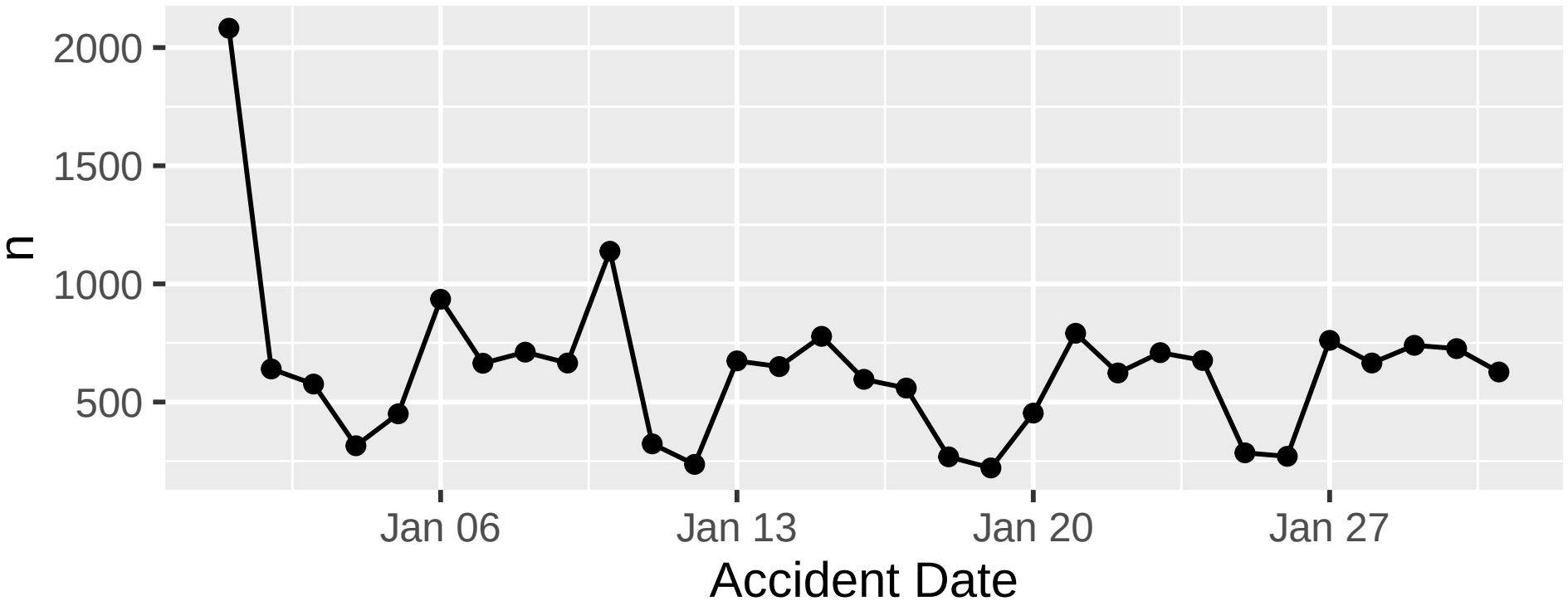
Zoom in on extreme value

```
wsddj <- ws %>% filter(year==2001&month=="Sep") %>%
  group_by(`Accident Date`) %>% tally()
ggplot(wsddj, aes(x=`Accident Date`, n)) +
  geom_point() + geom_line()
```



Compare with normal month

```
wsddj <- ws %>% filter(year==2014&month=="Jan") %>%
  group_by(`Accident Date`) %>% tally()
ggplot(wsddj, aes(x=`Accident Date`, n)) +
  geom_point() + geom_line()
```



Your turn

- Use your data wrangling skills to extract Sep 11, 2001 from the data
- Tabulate the claim types
- Is this what you expected?
- Brainstorm with your neighbour ways to investigate if these numbers are normal

Long time series

- Difficult to display long time series
- Points for counts has some flexibility to adjust to screen resolution
- Not aesthetically pleasing, aggregate at larger level

Some helper functions to compute time windows in months

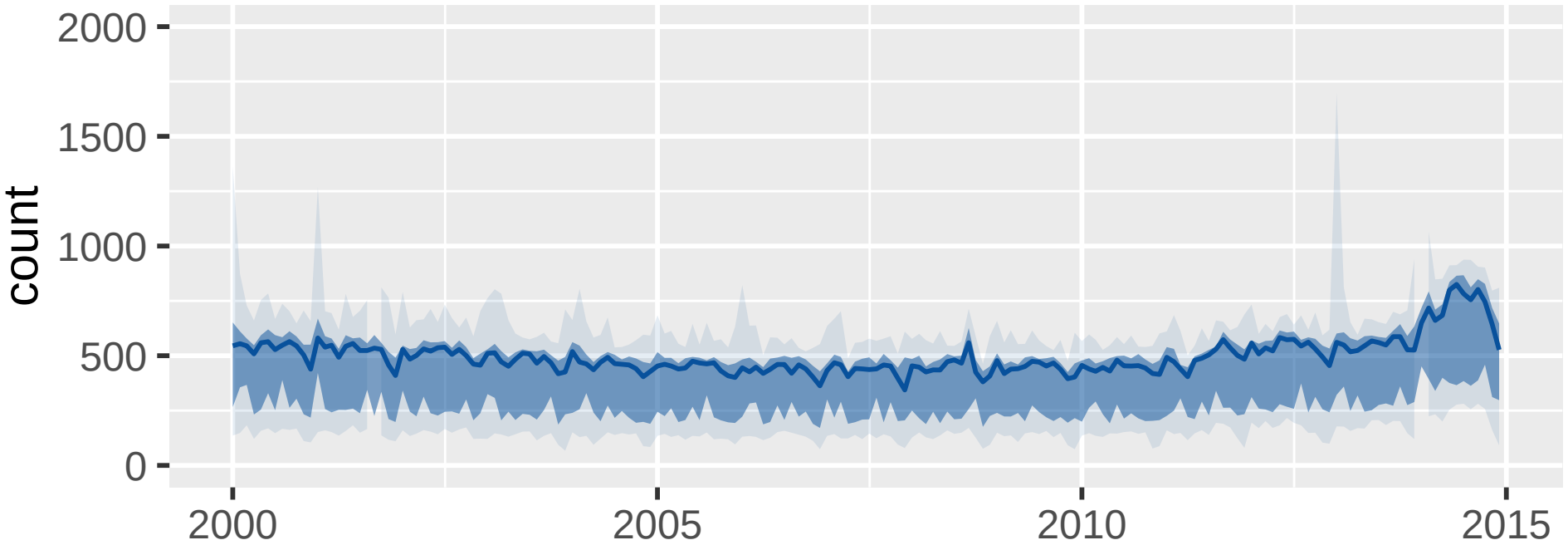
```
monnb <- function(d) {
  lt <- as.POSIXlt(as.Date(d, origin="1900-01-01"))
  lt$year*12 + lt$mon
}
mondf <- function(d1, d2) {
  monnb(d2) - monnb(d1)
}
```

Compute statistics by months: based on daily counts, what are the min, q1, median, q3, max for each month.

```
ws$timeindx_mths <- mondf(min(ws$`Accident Date`),
                          ws$`Accident Date`)
wsw <- ws %>% group_by(timeindx) %>%
  summarise(n=length(timeindx),
            timeindx_mths=timeindx_mths[1],
            date=min(`Accident Date`))
wsw.s <- wsw %>% group_by(timeindx_mths) %>%
  summarise(m=median(n), q1=quantile(n, 0.25),
            q3=quantile(n, 0.75), min=min(n),
            max=max(n), date=min(date))
```

Plot it

```
ggplot(wsw.s, aes(x=date, y=m)) +
  geom_ribbon(aes(ymin=min, ymax=max), fill="#08519C",
              alpha=0.1) +
  geom_ribbon(aes(ymin=q1, ymax=q3), fill="#08519C",
              alpha=0.5) +
  geom_line(aes(y=m), colour="#08519C") +
  ylim(c(0,2000)) +
  xlab("") + ylab("count")
```



Your turn

Create plots to:

- Examine the trend of claims by district. Is there a difference in overall trend?
- Examine the weekly pattern of claims by district. Are claims typically on week days everywhere?

Back to education - how to make the plots

Read the OECD PISA data

```
student2012.sub <- readRDS("data/student_sub.rds")
dim(student2012.sub)
#> [1] 271323     50
student2012.sub$ST04Q01 <- factor(student2012.sub$ST04Q01,
  levels=c(1,2), labels=c("Female", "Male"))
```

Calculate the statistics

```
student2012.stats <- student2012.sub %>%
  group_by(CNT) %>%
  summarise(wmathgap=weighted.mean(PV1MATH[ST04Q01=="Male"],
                    w=SENWGT_STU[ST04Q01=="Male"], na.rm=T)-
                  weighted.mean(PV1MATH[ST04Q01=="Female"],
                    w=SENWGT_STU[ST04Q01=="Female"], na.rm=T))
```

Plot these, check it works

```
ggplot(data=student2012.stats) +
  geom_point(aes(x=CNT, y=wmathgap), size=3) +
 coord_flip() + theme_bw()
```

Need to order!

Your turn

Use your wrangling skills to order the countries by size of difference

Helper functions to create bootstrap intervals for each mean difference

```
cifn <- function(d, i) {
  x <- d[i,]
  ci <- weighted.mean(x$PV1MATH[x$ST04Q01=="Male"],
          w=x$SENWGT_STU[x$ST04Q01=="Male"], na.rm=T)-
        weighted.mean(x$PV1MATH[x$ST04Q01=="Female"],
          w=x$SENWGT_STU[x$ST04Q01=="Female"], na.rm=T)
  ci
}
```

```
bootfn <- function(d) {
  r <- boot(d, statistic=cifn, R=100)
  l <- sort(r$t)[5]
  u <- sort(r$t)[95]
  ci <- c(l, u)
  return(ci)
}
```

Apply ci functions to data

```
student2012.sub.summary.gap.boot <- student2012.sub %>%
  split(.$CNT) %>% purrr::map(bootfn) %>% data.frame() %>%
  gather(CNT, value)
student2012.sub.summary.gap.boot$ci <-
  rep(c("ml","mu"),
      length(unique(student2012.sub.summary.gap.boot$CNT)))
student2012.sub.summary.gap.boot.wide <-
  student2012.sub.summary.gap.boot %>%
  spread(ci, value)
student2012.sub.summary.gap <- merge(student2012.stats,
  student2012.sub.summary.gap.boot.wide)
```

Match three digit codes to country names, more recognisable

```
student2012.sub.summary.gap$name <- NA
for (i in 1:length(student2012.sub.summary.gap$name))
  student2012.sub.summary.gap$name[i] <-
  isoToName(as.character(student2012.sub.summary.gap$CNT[i]))
# QCN is Shanghai, not whole of China -
# Don't know what country TAP is
student2012.sub.summary.gap$name[
  student2012.sub.summary.gap$CNT == "QCN"] <-
  isoToName("CHN")
student2012.sub.summary.gap$name[
  student2012.sub.summary.gap$CNT == "TAP"] <-
  "TAP"
```

Create categorical gap variable to indicate significance difference

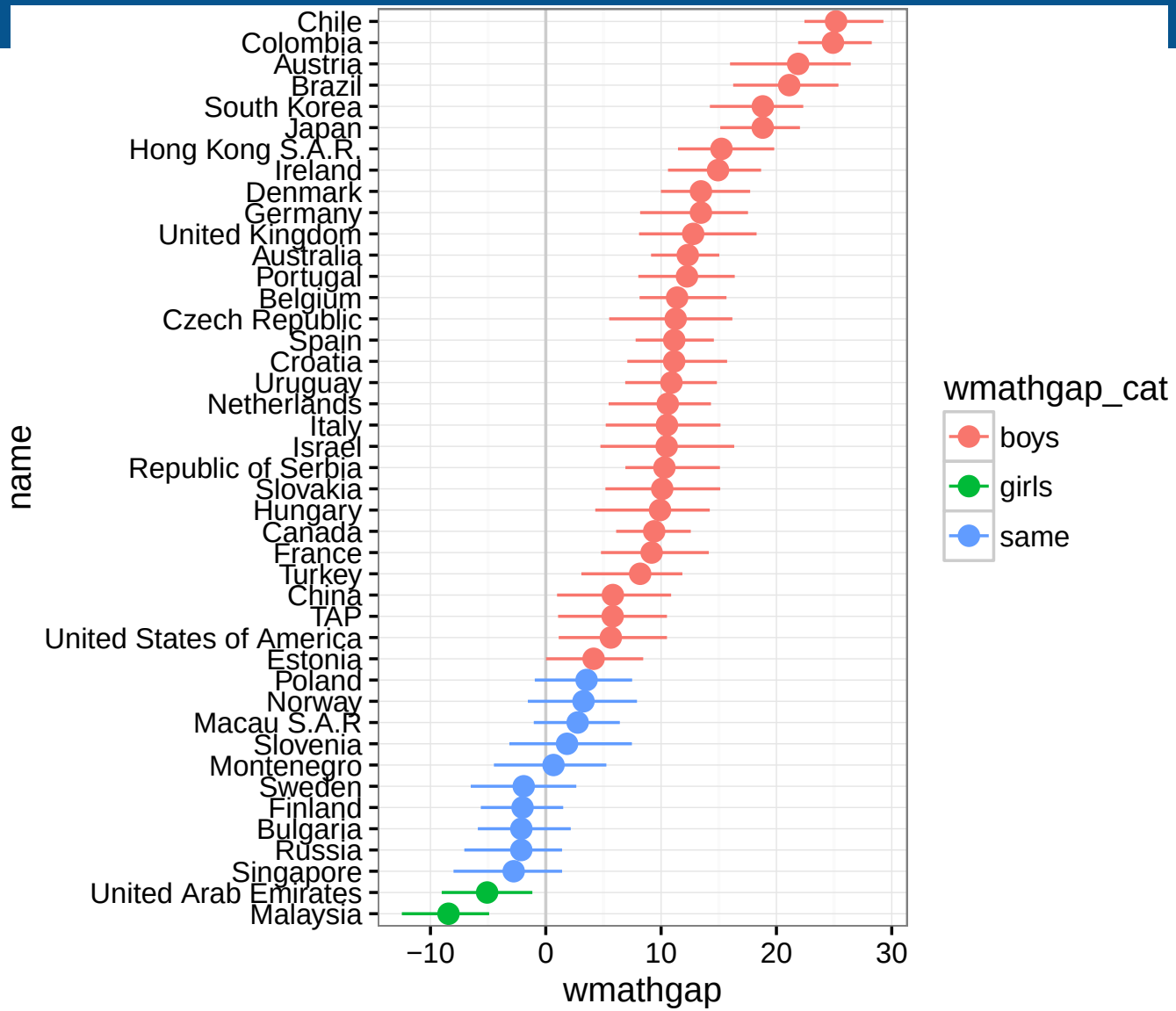
```
student2012.sub.summary.gap$wmathgap_cat <- "same"
student2012.sub.summary.gap$wmathgap_cat[
  student2012.sub.summary.gap$ml > 0] <- "boys"
student2012.sub.summary.gap$wmathgap_cat[
  student2012.sub.summary.gap$mu < 0] <- "girls"
```

Set order of countries by math gap

```
student2012.sub.summary.gap$CNT <- factor(
  student2012.sub.summary.gap$CNT,
      levels=student2012.sub.summary.gap$CNT[
        order(student2012.sub.summary.gap$wmathgap)])
student2012.sub.summary.gap$name <- factor(
  student2012.sub.summary.gap$name,
      levels=student2012.sub.summary.gap$name[
        order(student2012.sub.summary.gap$wmathgap)])
```

Plot it

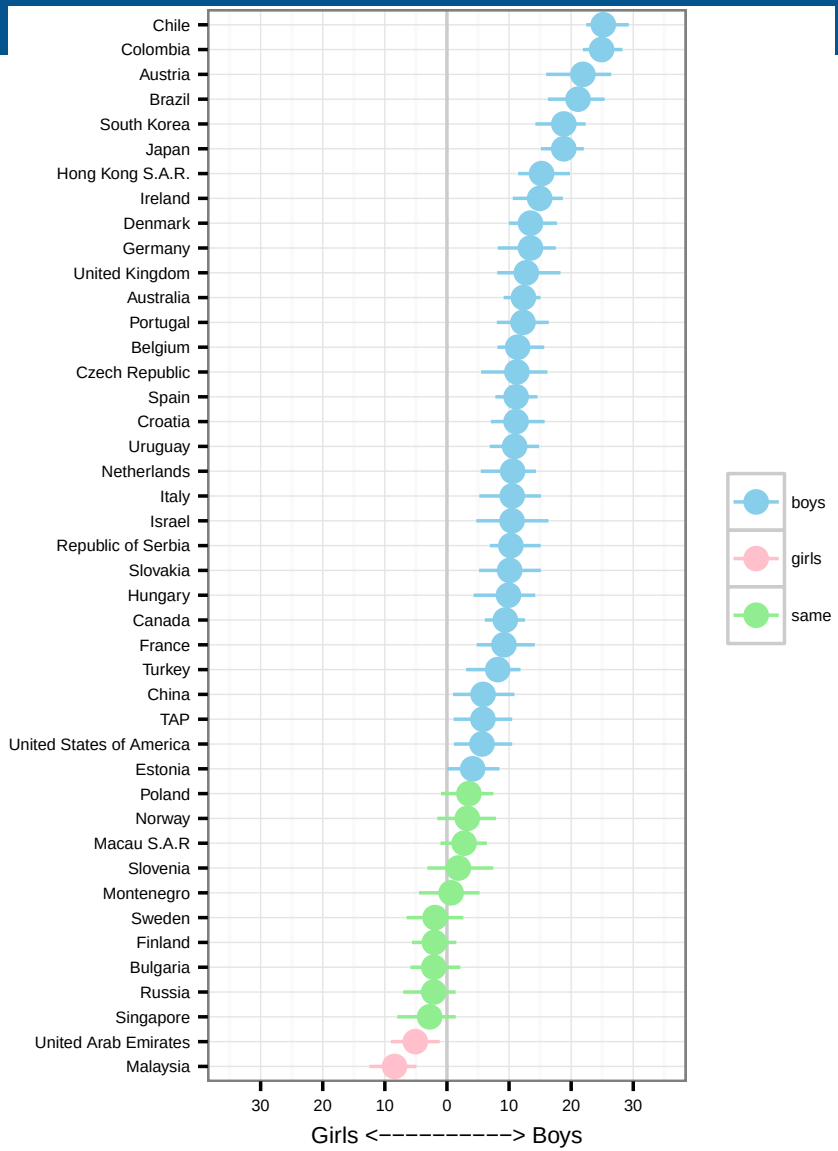
```
ggplot(data=student2012.sub.summary.gap) +
  geom_hline(yintercept=0, colour="grey80") +
  geom_point(aes(x=name, y=wmathgap, color=wmathgap_cat),
             size=3) +
  geom_segment(aes(x=name, xend=name, y=ml, yend=mu,
                   color=wmathgap_cat)) +
  coord_flip() + theme_bw()
```

Enhance the plot

- Labels
- Axis limits
- Grid lines
- Colour

```
ggplot(data=student2012.sub.summary.gap) +
  geom_hline(yintercept=0, colour="grey80") +
  geom_point(aes(x=name, y=wmathgap, color=wmathgap_cat), size
  geom_segment(aes(x=name, xend=name, y=ml, yend=mu,
     color=wmathgap_cat)) + xlab("") +
  scale_colour_manual("", values=c("boys"="skyblue",
    "girls"="pink", "same"="lightgreen")) +
  scale_y_continuous("Girls <----------> Boys",
    breaks=seq(-30, 30, 10), limits=c(-35, 35),
    labels=c(seq(30, 0, -10), seq(10, 30, 10))) +
  coord_flip() + theme_bw() +
  theme(axis.text.x = element_text(size=5),
        axis.text.y = element_text(size=5),
        axis.title = element_text(size=7),
        legend.text = element_text(size=5),
        legend.title = element_text(size=5))
```



Maps

Map data is essentially a set of points, and line segments. You can get maps from various sources, and wrangle the files/data into an R object. This can be merged with data to provide spatial context to problems.

```
world <- getMap(resolution = "low")
extractPolys <- function(p) {
  polys <- NULL
  for (i in 1:length(p)) {
    for (j in 1:length(p[[i]]@Polygons)) {
      x <- p[[i]]@Polygons[[j]]@coords
      polys$lon <- c(polys$lon, x[,1])
      polys$lat <- c(polys$lat, x[,2])
      polys$ID <- c(polys$ID, rep(p[[i]]@ID, nrow(x)))
      polys$region <- c(polys$region,
        rep(paste(p[[i]]@ID, j, sep="_"), nrow(x)))
      polys$order <- c(polys$order, 1:nrow(x))
    }
```

Here is what is looks like:

```
kable(head(polys))
```

lon	lat	ID	region	order
-70	12	Aruba	Aruba_1	1
-70	12	Aruba	Aruba_1	2
-70	12	Aruba	Aruba_1	3
-70	12	Aruba	Aruba_1	4
-70	13	Aruba	Aruba_1	5
-70	13	Aruba	Aruba_1	6

Join education data with map polygons

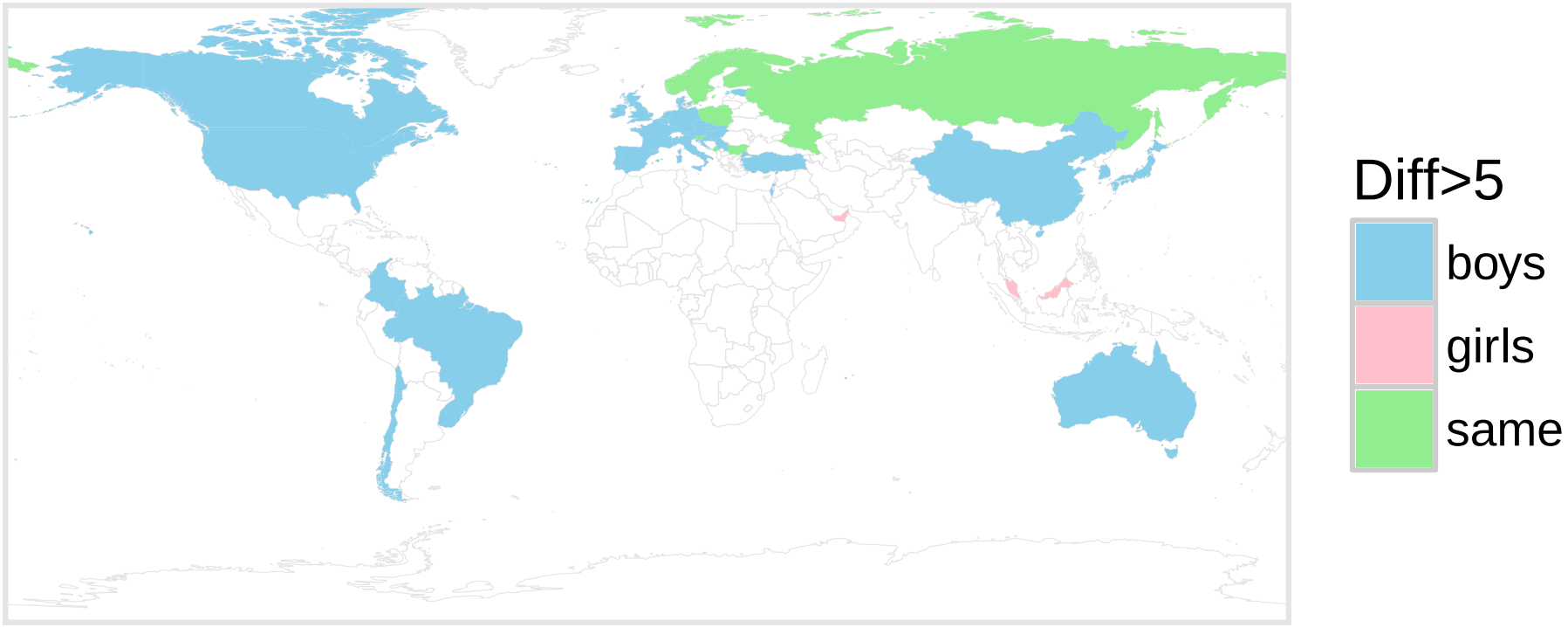
```
polys <- polys %>% rename(name = ID)
student2012.sub.map <- left_join(
  student2012.sub.summary.gap, polys)
student2012.sub.map <- student2012.sub.map %>%
  arrange(region, order)
```

Map theme

Make it look like a map, by tweaking the plot appearance

```
theme_map <- theme_bw()
theme_map$line <- element_blank()
theme_map$strip.text <- element_blank()
theme_map$axis.text <- element_blank()
theme_map$plot.title <- element_blank()
theme_map$axis.title <- element_blank()
theme_map$panel.border <- element_rect(
  colour = "grey90", size=1, fill=NA)
```

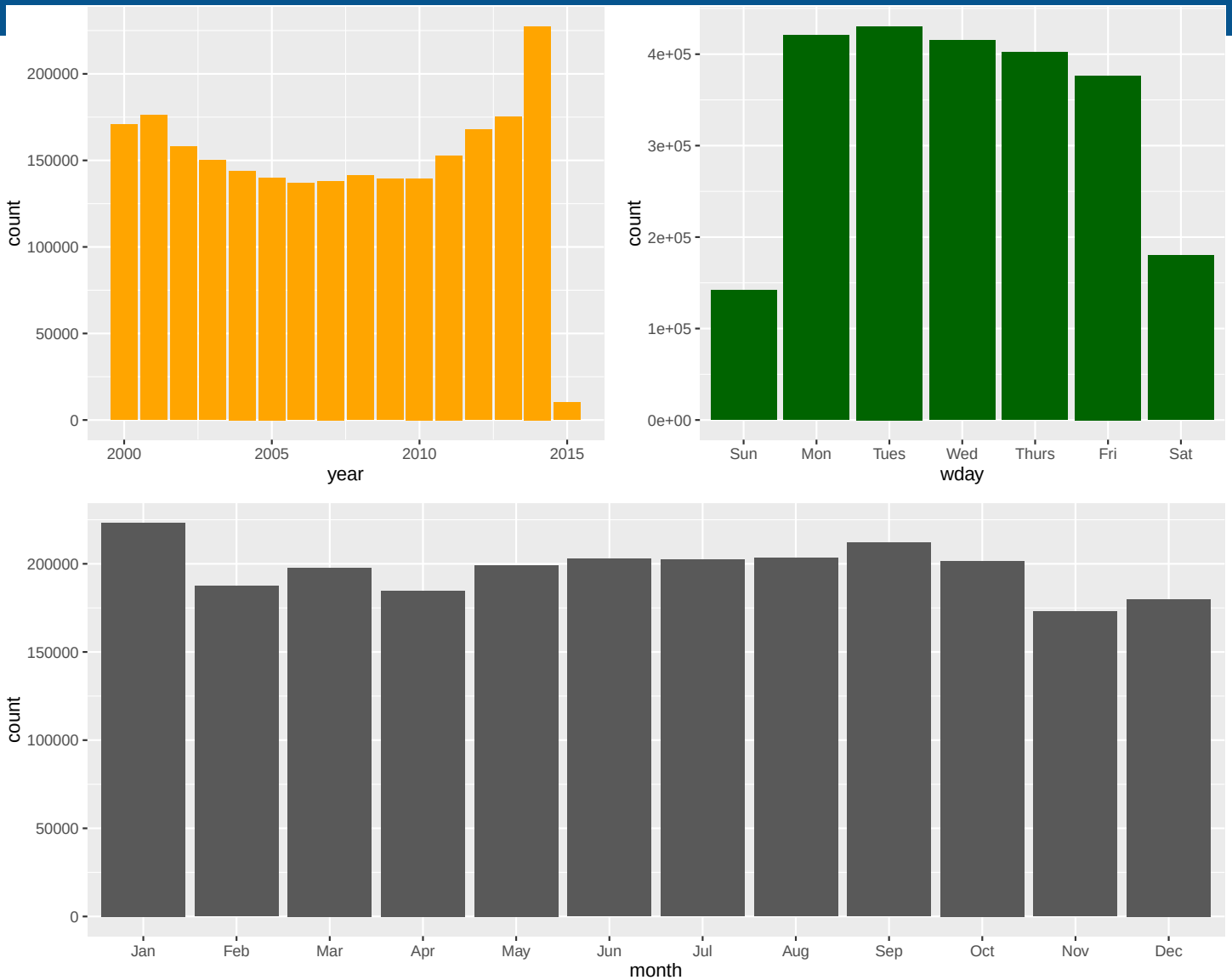
```
ggplot(data=polys) +
  geom_path(aes(x=lon, y=lat, group=region, order=order),
            colour=I("grey90"), size=0.1) +
  geom_polygon(data=student2012.sub.map, aes(x=lon, y=lat,
            group=region, order=order,
            fill=wmathgap_cat)) +
  scale_fill_manual("Diff>5", values=c("boys"="skyblue",
                                       "girls"="pink",
                                       "same"="lightgreen")) +
  scale_x_continuous(expand=c(0,0)) +
  scale_y_continuous(expand=c(0,0)) +
  coord_equal() + theme_map
```



Multiple plots on a sheet

Occasionally you would like to organise your plots in special ways. The `gridExtra` can be used to take individual plots and lay them out together.

```
ws <- workers %>% filter(year > 1999)
p1 <- ggplot(ws, aes(x=year)) + geom_bar()
p2 <- ggplot(ws, aes(x=wday)) + geom_bar()
p3 <- ggplot(ws, aes(x=month)) + geom_bar()
grid.arrange(p1, p2, p3, layout_matrix = rbind(c(1,2),c(3,3)))
```



Your turn

For your own data, or the NYC workers compensation data

- Determine a couple of questions to ask
- Write the code to compute the necessary quantities
- Make a plot (or plots) that helps to answer each of the question
- Compile this into a markdown document, and make it into a word file
- Show the instructors

Credits

Notes prepared by Di Cook, building on joint workshops with Carson Sievert, Heike Hofmann, Eric Hare, Hadley Wickham.