Reproducible hockey analysis using R Statistical Software

Michael Lopez, Skidmore College

Why are you here?

Is this you? You are a `Ctrl C` and then a `Ctrl V` person

Source: Yihui Xie

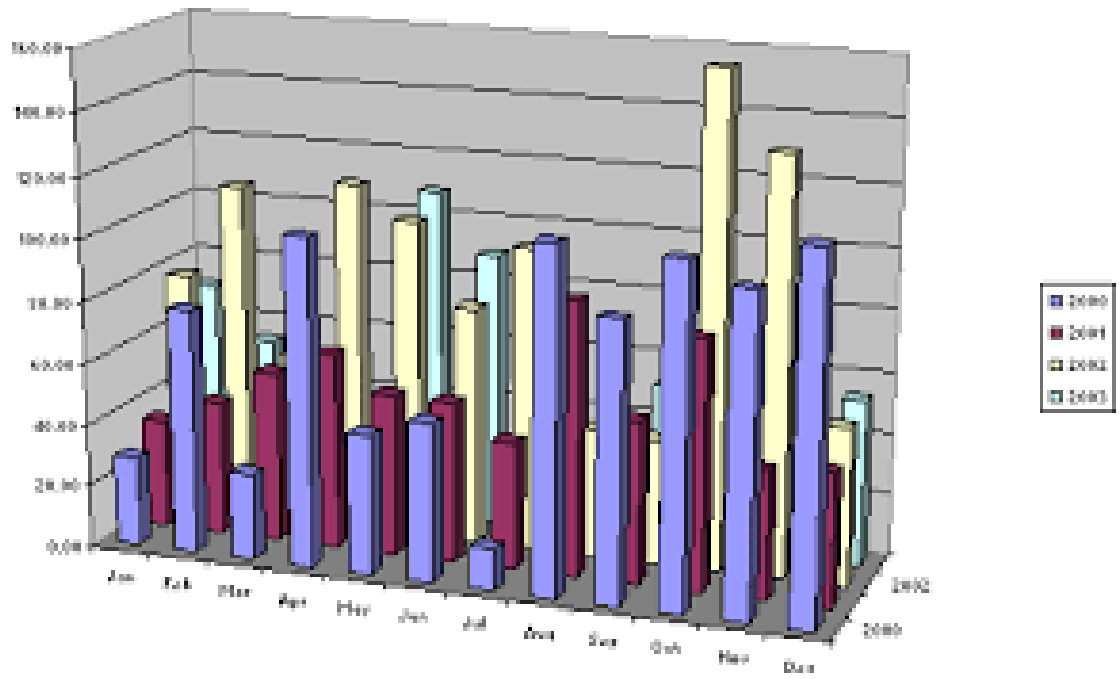
Why are you here?

Or maybe this is you? You are a Microsoft Excel programmer

	A	B	C
1	234	56.9	70.45
2	34.23	#DIV/0!	0.028
3	#NAME?	56%	705
4	0.51	72.04	#REF!
5	#NAME?	#VALUE!	950
6	386	67.89	#DIV/0!
7	90.45	650	217
8	702	734.967	32.967

Why are you here?

Or maybe this is you? You are a Microsoft Excel grapher

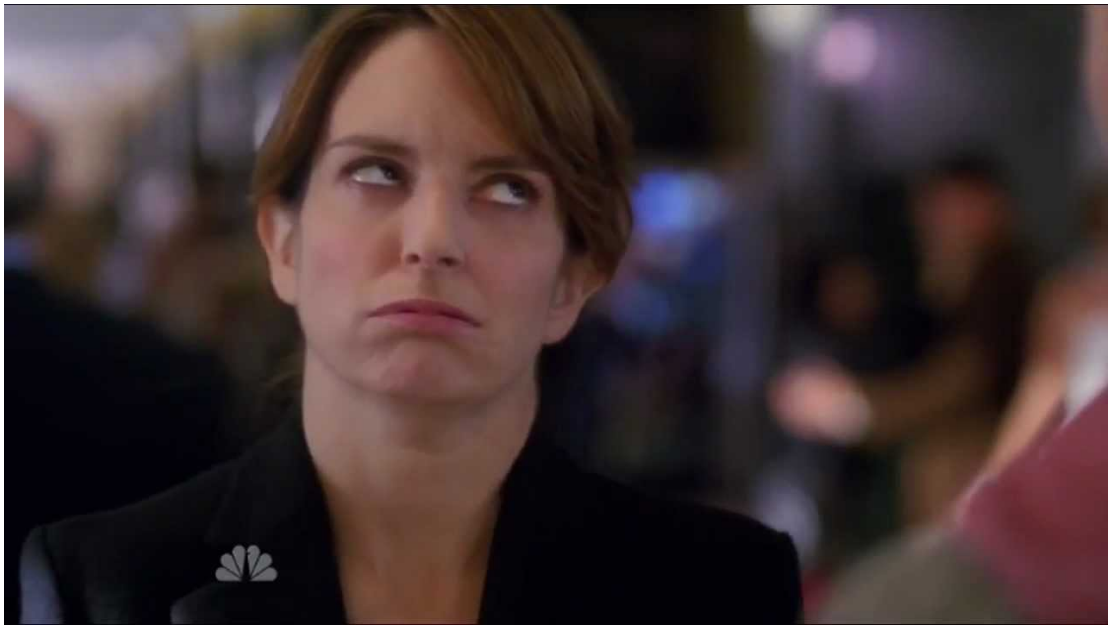


And what do you when. . . ?

You here these after finishing a project

- "The axis is too small"
- "Can you use assists per 60 instead of assists?"
- "We made a mistake in the data"
- "I want to see 5 v 5 rates only"
- "But you didn't adjust for. . . "

How do you react?



Of course. . .

This is still you. You are still a `Ctrl C` and then a `Ctrl V` person . . . so it's `Ctrl C` and then `Ctrl V` until infinity

So why are we here?

4 goals of today's workshop

1. R for reproducibility - no more copy and paste!
2. R for (improved) data visualization
3. R for (easier) data manipulation
4. Learn & have fun

Preliminary info

- R (link) and RStudio (link) are free to download
- Talk is available online at my website.
- Copy and paste the code! That's okay for today as you are just getting started
- Slides produced using Markdown, using code at on Github
- Sports and stats course at my website

What is reproduciblility?

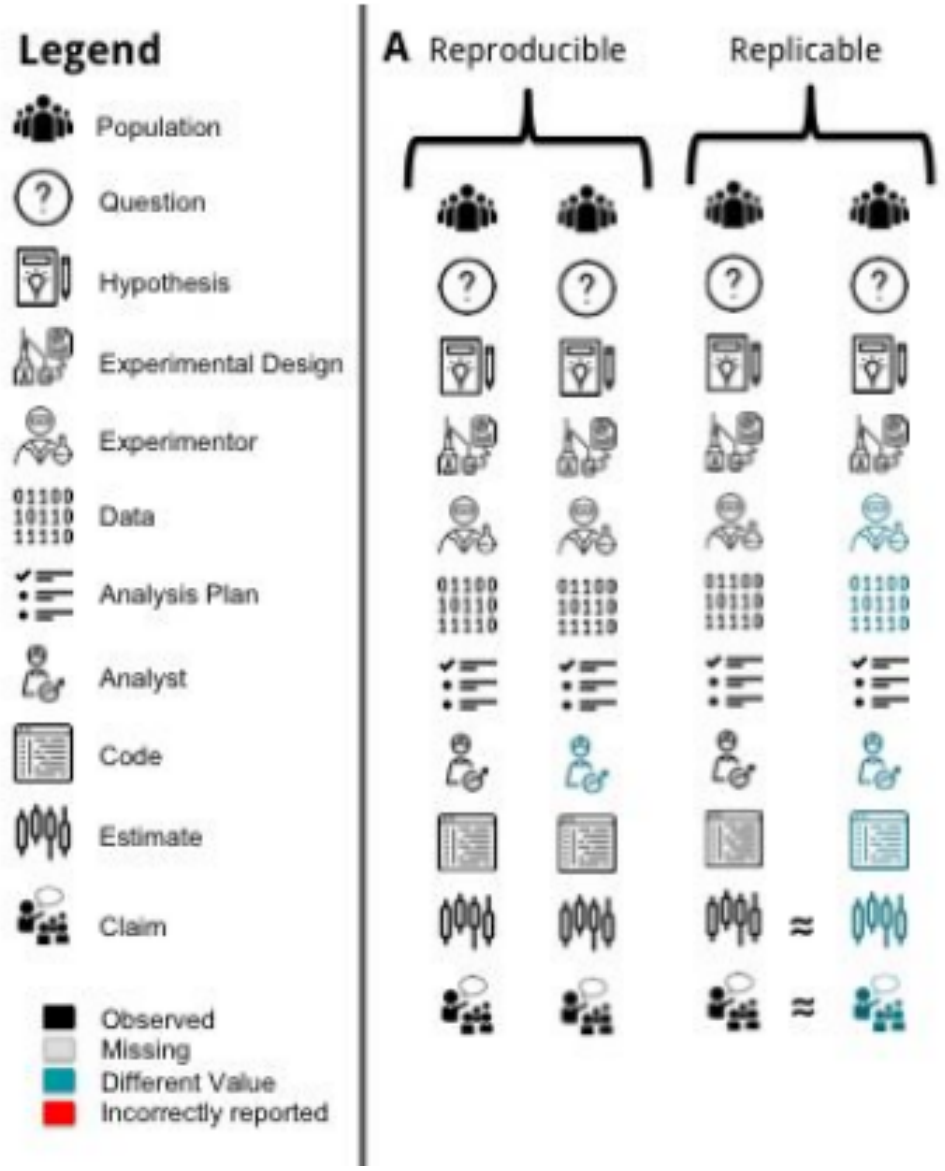


Figure 1: Reproducibility v. Replicability via Patil et al (link)

Why reproduciblility?

- Focus on content
- Verification of findings
- Increased citations
- Field advances quicker
- Quicker comparisons of novel approaches
- Consistent workflow, easier edits

Mistakes and mishaps of non-reproducible research

"Divorce study felled by a coding error gets a second chance"

"Excel hell messes up ~20 per cent of genetic science papers"

"Scientists replicated 100 recent psychology experiments. More than half of them failed"

Reproducibility in hockey

- nhlscrapr and WAR on Ice
- Shootout study via Lopez and Schuckers (link)
- Corsica
- What else is there? Seemingly, not much

Reproducibility flow

1. Read in data
2. Manipulate data
3. Analysis & Visualization
4. Share!

Preliminary info, R

Creating a Markdown document

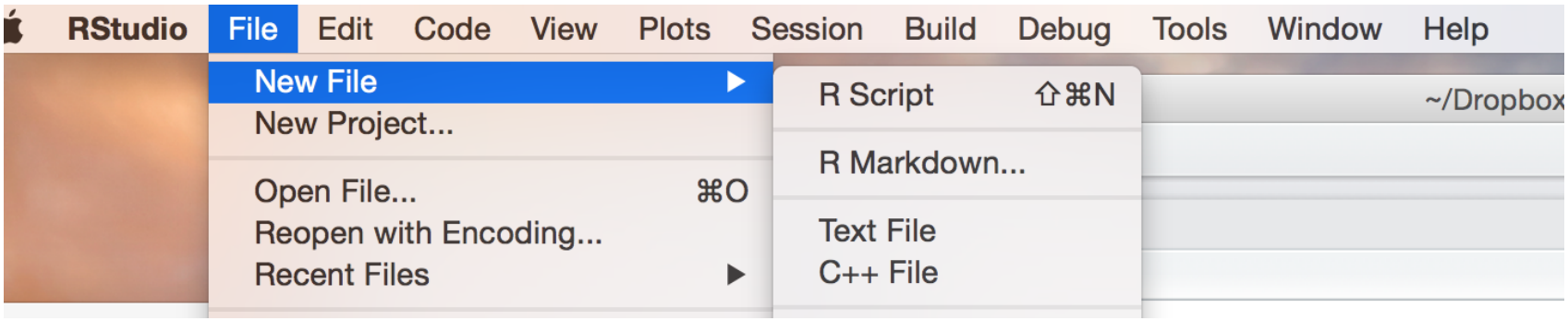


Figure 2:

Preliminary info, R

Packages in R for the sports aficionado

1. `dplyr`, `tidyr` for data manipulation
2. `stringr` for pesky characters
3. `lubridate` for dates
4. `ggplot2` for visualization
5. `lme4`, `mgcv`, `rjags` for advanced modeling
6. `glmnet`, `randomForest`, `caret` for machine learning
7. `RMarkdown` for reproducibility
8. `readr`, `googlesheets` for uploading data
9. `rvest`, `XML` for scraping data off web
10. `BradleyTerry2` for team rankings

```
install.packages("dplyr")
```

Step 1: Read in data

Datasets for immediate use (click on each for hyperlinks)

- Hockey reference
- War-on-ice raw data, Ventura/Thomas
- Passing project, Stimson
- Player data, Vollman
- Individual blogs w/ downloadable data
- Not NHL.com

Step 1: Read in data

Example: Scraiping game outcomes from hockey-reference

```
library(stringr); library(XML); library(BradleyTerry2)
library(dplyr); library(ggplot2); options(dplyr.width = Inf)
url <- c("http://www.hockey-reference.com/leagues/NHL_2016_games.html")
nhl.df <- readHTMLTable(url)
reg.season <- nhl.df$games
reg.season %>%
  head(3)
```

```
##         Date            Visitor G                Home G    Att.  LOG Notes
## 1 2015-10-07 Vancouver Canucks 5     Calgary Flames 1  19,289 2:32
## 2 2015-10-07  New York Rangers 3 Chicago Blackhawks 2  22,104 2:28
## 3 2015-10-07   San Jose Sharks 5  Los Angeles Kings 1  18,230 2:40
```

Step 1: Read in data

Example: Play-by-play data via `nhlscrapr` package

```
load(url("http://war-on-ice.com/data/nhlscrapr-20142015.RData"))
grand.data %>% head(3)
```

```
##     season gcode refdate event period seconds  etype   a1   a2   a3   a4
## 1 20142015 20001    4663     1      1     0.0    FAC 1157 3917 3876 1371
## 2 20142015 20001    4663     2      1    19.0   MISS 1157 3917 3876 1371
## 3 20142015 20001    4663     3      1    27.5 CHANGE 1157 3917 3876 1371
##     a5 a6  h1   h2   h3   h4   h5 h6 ev.team ev.player.1 ev.player.2
## 1 5043  1 561 2514 5385 4050 4366  1     MTL        1157         561
## 2 5043  1 561 2514 5385 4050 4366  1     MTL        1157           1
## 3 5043  1 561 2514 5385 4050 4366  1                   1           1
##   ev.player.3 distance  type homezone xcoord ycoord awayteam hometeam
## 1           1       NA           Neu     NA     NA      MTL      TOR
## 2           1       41 Wrist     Def    -52     31      MTL      TOR
## 3           1       NA           Neu     NA     NA      MTL      TOR
##   home.score away.score event.length away.G home.G home.skaters
## 1          0          0          0.0   4169    359            6
## 2          0          0         19.0   4169    359            6
## 3          0          0          8.5   4169    359            6
##   away.skaters adjusted.distance shot.feature import.ies loc.section
## 1            6                NA                     0           0
## 2            6          37.43114                     1           0
## 3            6                NA                     0           0
##   new.loc.section newxc newyc score.diff.cat subdistance
## 1               0    NA    NA              3          NA
## 2               4    55   -28              3           9
## 3               0    NA    NA              3          NA
```

Step 1: Read in data

Example: Roster data via `nhlscrapr` package

```
load(url("http://war-on-ice.com/data/nhlscrapr-core.RData"))
roster.unique %>% head(2)
```

```
## Source: local data frame [2 x 17]
## Groups: player.id [2]
##
##     pos  last first      numfirstlast   firstlast index player.id
##   <chr> <chr> <chr>             <chr>       <chr> <dbl>     <dbl>
## 1  <NA>                                             1         1
## 2     C Aalto Antti [TSN] Antti Aalto Antti Aalto     2         2
##      woi.id    pC    pL    pR    pD    pG        DOB Height Weight Shoots
##       <chr> <dbl> <dbl> <dbl> <dbl> <dbl>      <chr>  <chr>  <chr>  <chr>
## 1 xxxxxxxNA     0     0     0     0     0       <NA>   <NA>   <NA>   <NA>
## 2 aaltoan75     0     0     0     0     0 1975-03-04    6-1    210      L
```

Step 2: Data Munging

The `dplyr` package aims to provide a function for each basic verb of data manipulation:

- `filter()` (and `slice()`) for finding specific rows and columns
- `select()` (and `rename()`) for simplying a data frame
- `mutate()` for creating a new variable
- `arrange()` for ordering
- `summarise()` for means, medians, max, min, etc
- `sample_n()` for random samples
- `group_by()` for within-group computation
- `left_join()` (and `inner_join`, `right_join`) for merging

Data munging, hockey play-by-play

Example: munging with the `dplyr` package

```
grand.data %>%
  filter(etype == "GOAL") %>%
  inner_join(roster.unique, by = c("ev.player.1"="player.id")) %>%
  select(gcode, ev.team, ev.player.1, firstlast,
         distance, ev.team, hometeam, awayteam, seconds) %>%
  sample_n(3)
```

```
##      gcode ev.team ev.player.1        firstlast distance hometeam awayteam
## 5580 20983     OTT         832    Alex Chiasson        8      OTT      CGY
## 977  20176     CAR        3633       Riley Nash       24      CBJ      CAR
## 4943 20869     STL        4066 Alex Pietrangelo       31      STL      BOS
##      seconds
## 5580     734
## 977     1028
## 4943    1456
```

Data munging, hockey play-by-play

Example: munging Corsi with the `dplyr` package

```
brad <- grand.data %>%
  filter(a1 == 3190 | a2 == 3190 | a3 == 3190 |
         h1 == 3190 | h2 == 3190 | h3 == 3190,
         away.skaters == 6, home.skaters == 6,
         etype == "SHOT" | etype == "GOAL") %>%
  mutate(bos.event = as.numeric((ev.team == "BOS")),
         corsi.count = 1*bos.event + -1*(1-bos.event),
         cum.corsi = cumsum(corsi.count),
         index = 1:n())
brad %>%
  select(season, gcode, period, ev.team, bos.event, corsi.count, cum.corsi, index) %>%
  head(6)
```

```
##     season gcode period ev.team bos.event corsi.count cum.corsi index
## 1 20142015 20002      1     BOS         1           1         1     1
## 2 20142015 20002      1     BOS         1           1         2     2
## 3 20142015 20002      1     BOS         1           1         3     3
## 4 20142015 20002      1     PHI         0          -1         2     4
## 5 20142015 20002      2     PHI         0          -1         1     5
## 6 20142015 20002      2     BOS         1           1         2     6
```

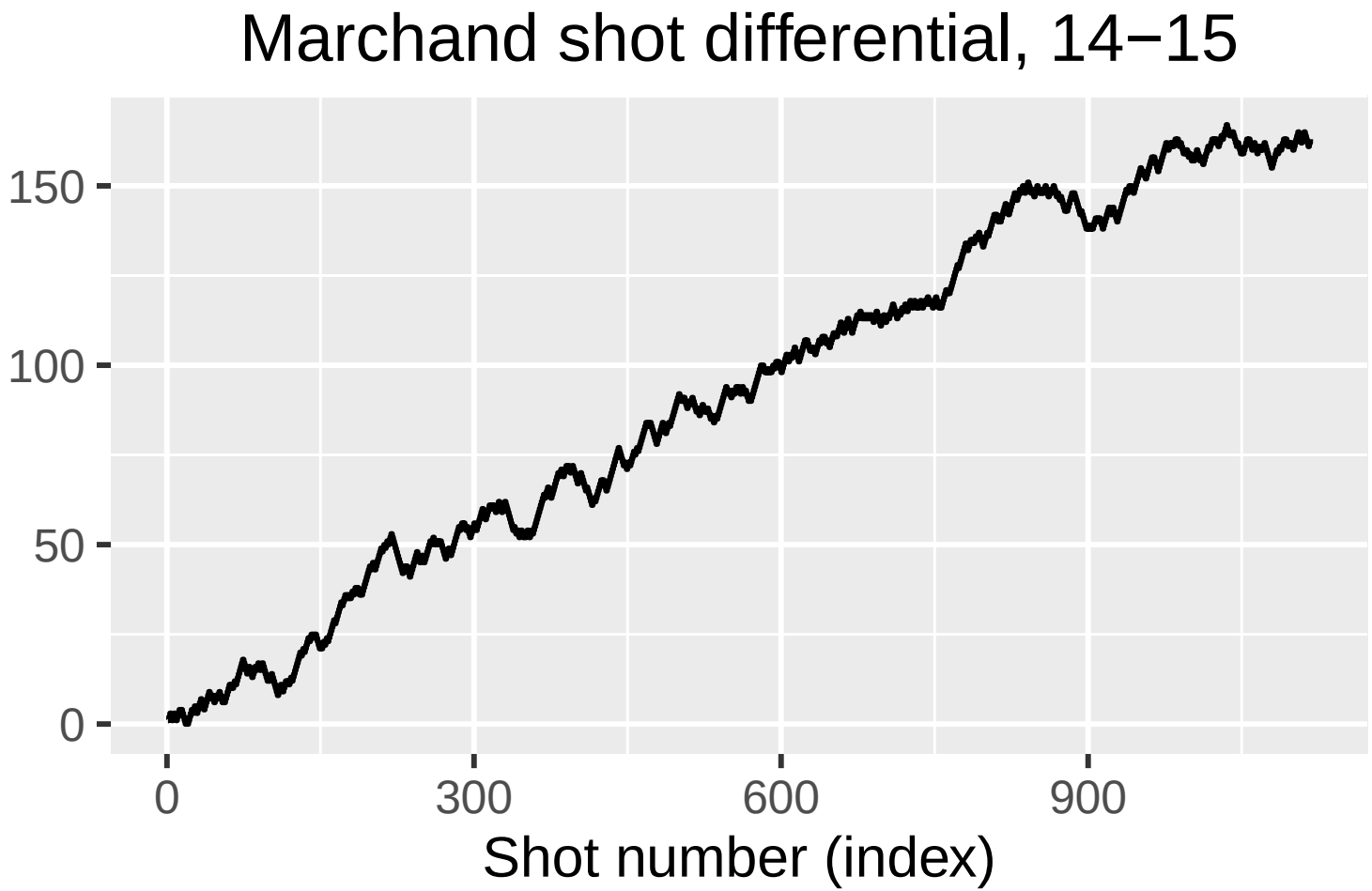
Step 3: Visualizing data

The `ggplot2`: seven compon grammar of graphics

- `data` for data set
- `aes` for plot aesthetics (variable choices)
- `layers()` for coloring, size, etc
- `scales` for identification
- `coordinates` for coordinate system
- `faceting` for distinct plots
- `theme` for extras and background

Wrangling and visualization, hockey play-by-play

```
ggplot(data = brad, aes(x = index, y=cum.corsi)) +
      geom_step() +
      xlab("Shot number (index)") +
      ylab("") +
      ggtitle("Marchand shot differential, 14-15")
```

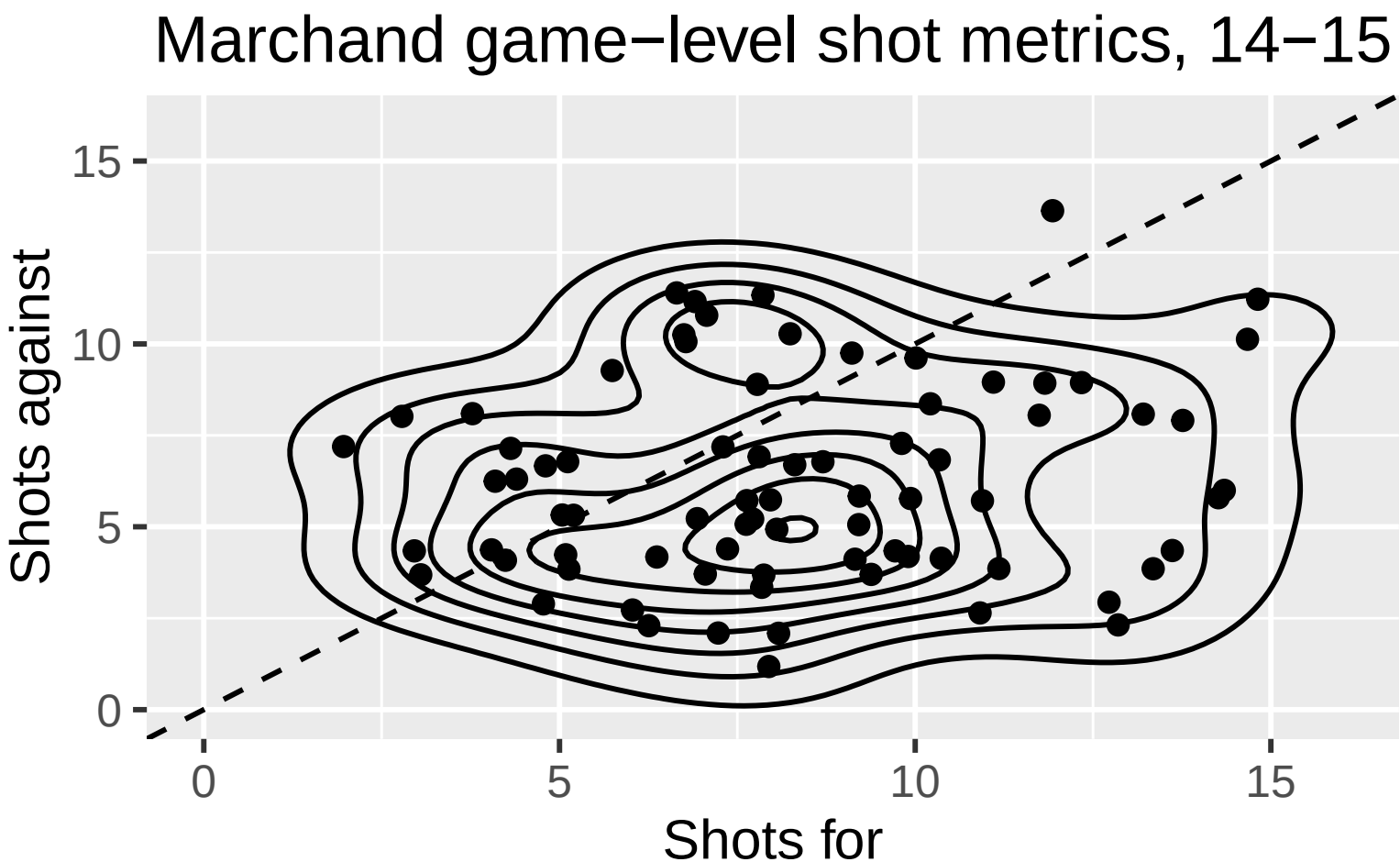


Wrangling and visualization, hockey play-by-play

```
brad.game <- grand.data %>%
  filter(a1 == 3190 | a2 == 3190 | a3 == 3190 |
         h1 == 3190 | h2 == 3190 | h3 == 3190,
         away.skaters == 6, home.skaters == 6,
         etype == "SHOT" | etype == "GOAL") %>%
  mutate(bos.event = as.numeric((ev.team == "BOS"))) %>%
  group_by(gcode) %>%
  summarise(corsi.game.for = sum(bos.event),
            corsi.game.against = sum(!bos.event),
            corsi.game.diff = corsi.game.for - corsi.game.against)
```

Wrangling and visualization, hockey play-by-play

```
p <- ggplot(brad.game, aes(corsi.game.for, corsi.game.against)) +
  geom_jitter(colour = "black") + geom_density2d(colour = "black") +
  scale_x_continuous(lim = c(0, 16)) +
  scale_y_continuous(lim = c(0, 16)) +
      xlab("Shots for") +
      ylab("Shots against") +
  geom_abline(intercept = 0, slope = 1, linetype = 2)
p + ggtitle("Marchand game-level shot metrics, 14-15")
```



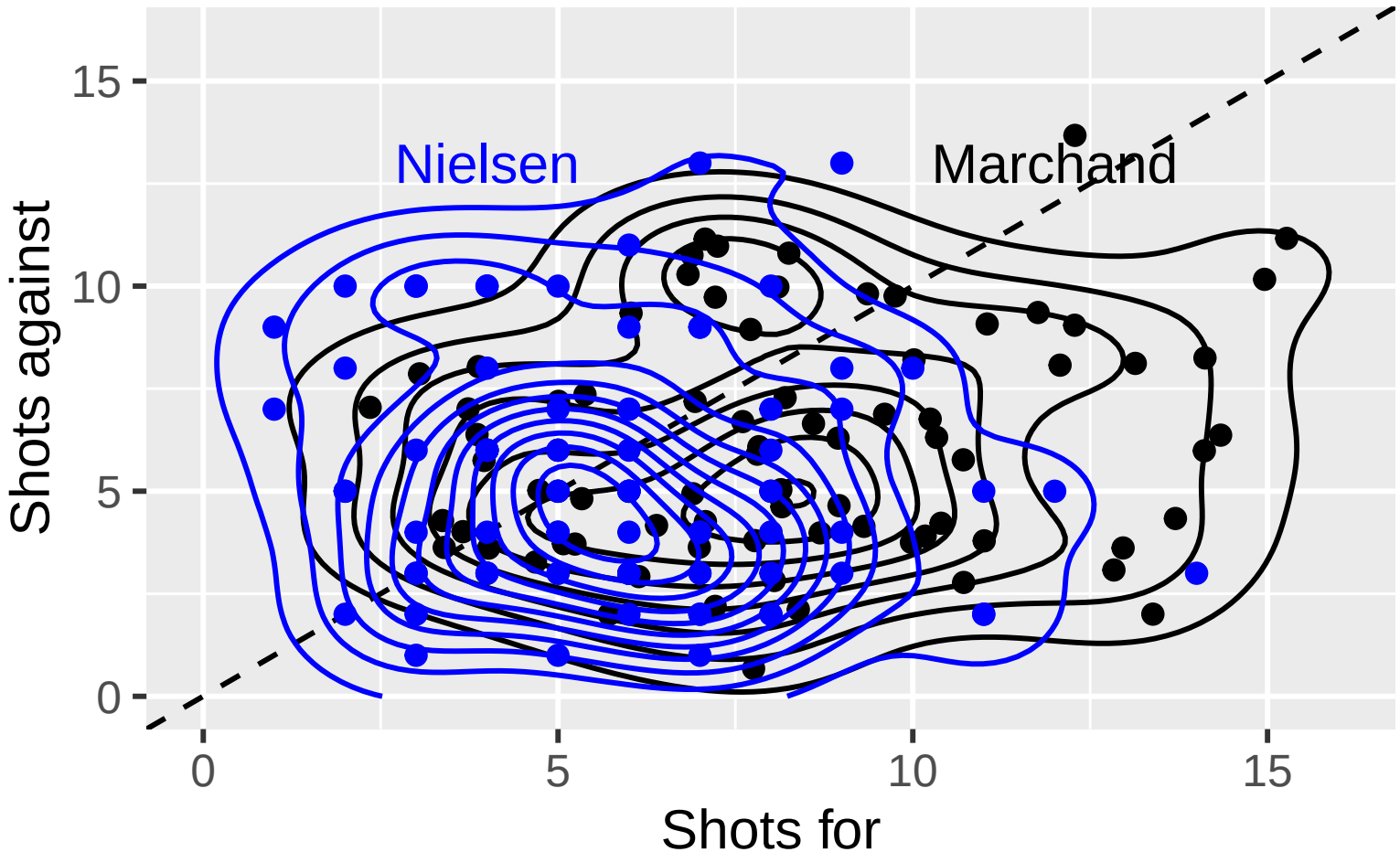
Functions, Wrangling & Visualization

```
player.game <- function(id, team){
game.data <- grand.data %>%
  filter(a1 == id | a2 == id | a3 == id |
         h1 == id | h2 == id | h3 == id,
         away.skaters == 6, home.skaters == 6,
         etype == "SHOT" | etype == "GOAL") %>%
  mutate(team.event = as.numeric((ev.team == team))) %>%
  group_by(gcode) %>%
  summarise(corsi.game.for = sum(team.event),
            corsi.game.against = sum(!team.event),
            corsi.game.diff = corsi.game.for - corsi.game.against)
return(game.data)
}

nielsen.game <- player.game(3690, "NYI")
```

Marchand vs. Nielsen, game-level data

```
p +  geom_density2d(data = nielsen.game, aes(corsi.game.for, corsi.game.against),
                    colour = "blue") +
  geom_point(data = nielsen.game, aes(corsi.game.for, corsi.game.against),
                    colour = "blue") +
  annotate("text", x = 4, y = 13, label = "Nielsen", colour = "blue") +
  annotate("text", x = 12, y = 13, label = "Marchand")
```



Step 3: R for statistical modeling

Team comparisons, 2015-16

```
names(reg.season)[3] <- "vis.goals"
names(reg.season)[5] <- "home.goals"

reg.season1 <- reg.season %>%
  select(Date, Visitor, vis.goals, Home, home.goals) %>%
    mutate(goals.diff = as.numeric(as.character(home.goals)) -
                        as.numeric(as.character(vis.goals)),
           home.win = as.numeric(goals.diff > 0))
reg.season1 %>% head(3)
```

```
##         Date           Visitor vis.goals               Home home.goals
## 1 2015-10-07 Vancouver Canucks         5     Calgary Flames          1
## 2 2015-10-07  New York Rangers         3 Chicago Blackhawks          2
## 3 2015-10-07   San Jose Sharks         5  Los Angeles Kings          1
##   goals.diff home.win
## 1         -4        0
## 2         -1        0
## 3         -4        0
```

Step 3: R for statistical modeling

Team comparisons, 2015-16

```
homeBT <- BTm(outcome = home.win,
      player1 = data.frame(team = Home, home.ice = 1),
      player2 = data.frame(team = Visitor, home.ice = 0),
        ~ team + home.ice,
        id = "team", data = reg.season1)

abilities <- data.frame(BTabilities(homeBT))
abilities <- abilities %>%
  mutate(ability = ability - mean(ability))
abilities$Team <- word(rownames(BTabilities(homeBT)),-1)
abilities %>% head(3)
```

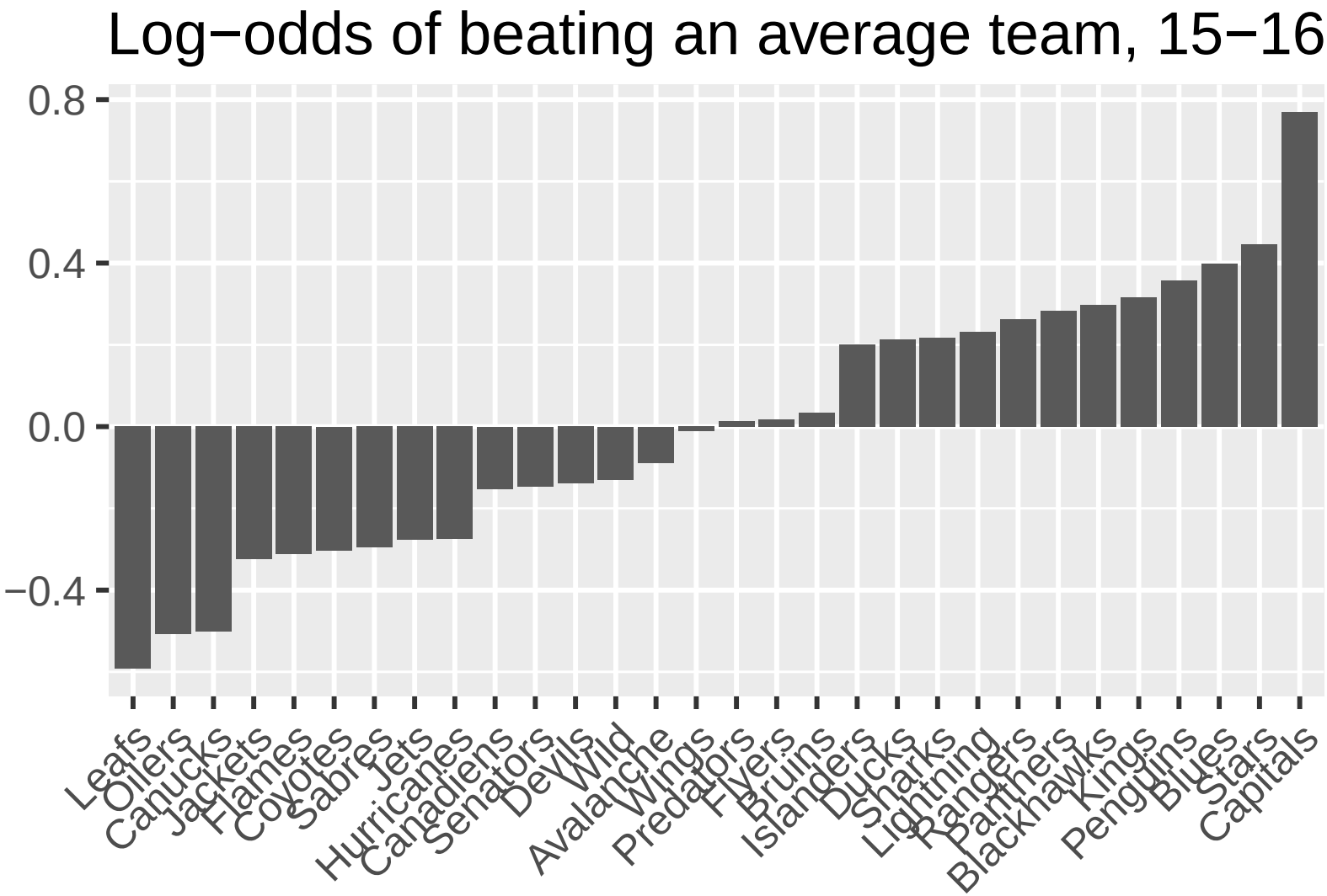
```
##       ability      s.e.    Team
## 1  0.21254247 0.0000000   Ducks
## 2 -0.30289146 0.3109341 Coyotes
## 3  0.03414039 0.3159320  Bruins
```

Step 3: R for statistical modeling

Visualizing team abilities

```
abilities$Team <-factor(abilities$Team,
                        levels= abilities[order(abilities$ability),"Team"])
p<- ggplot(abilities, aes(x = Team, y = ability)) +
  geom_bar(stat = "identity")  +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Log-odds of beating an average team, 15-16")+ xlab("") + ylab("")
```

Step 3: R for statistical modeling

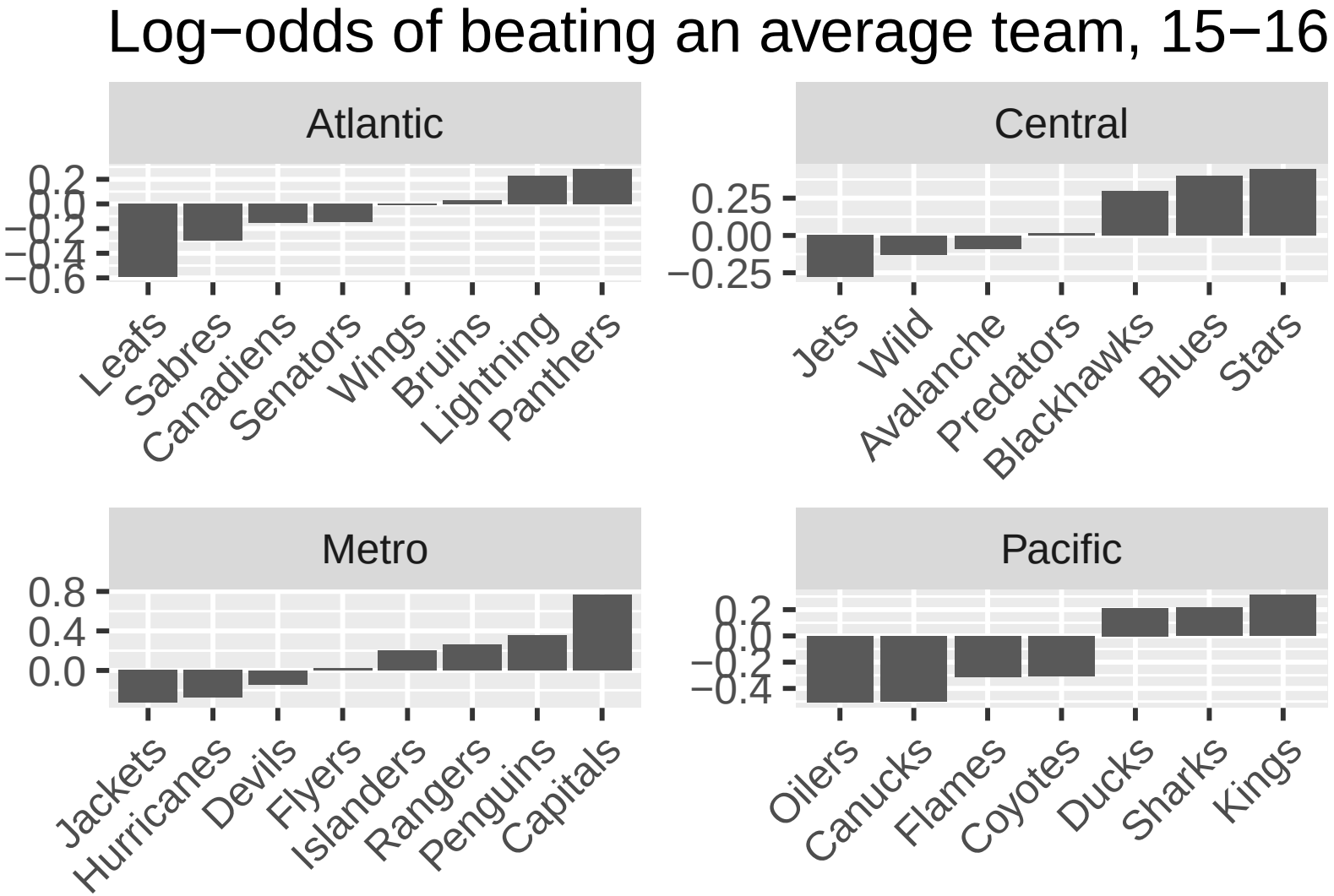


Step 3: R for statistical modeling

```
div <- c("Metro", "Atlantic", "Central", "Pacific")
div.teams <- c(div[4], div[4], div[2], div[2], div[4],
               div[1], div[3], div[3], div[1], div[3],
               div[2], div[4], div[2], div[4], div[3],
               div[2], div[3], div[1], div[1], div[1],
               div[2], div[1], div[1], div[4], div[3],
               div[2], div[2], div[4], div[1], div[3])
abilities$division <- div.teams

p <- ggplot(abilities, aes(x = Team, y = ability)) +
  geom_bar(stat = "identity")  +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Log-odds of beating an average team, 15-16")+ xlab("") + ylab("") +
  facet_wrap(~division, scales = "free")
```

R for statistical modeling



R for advanced statistical modeling

Project in the works

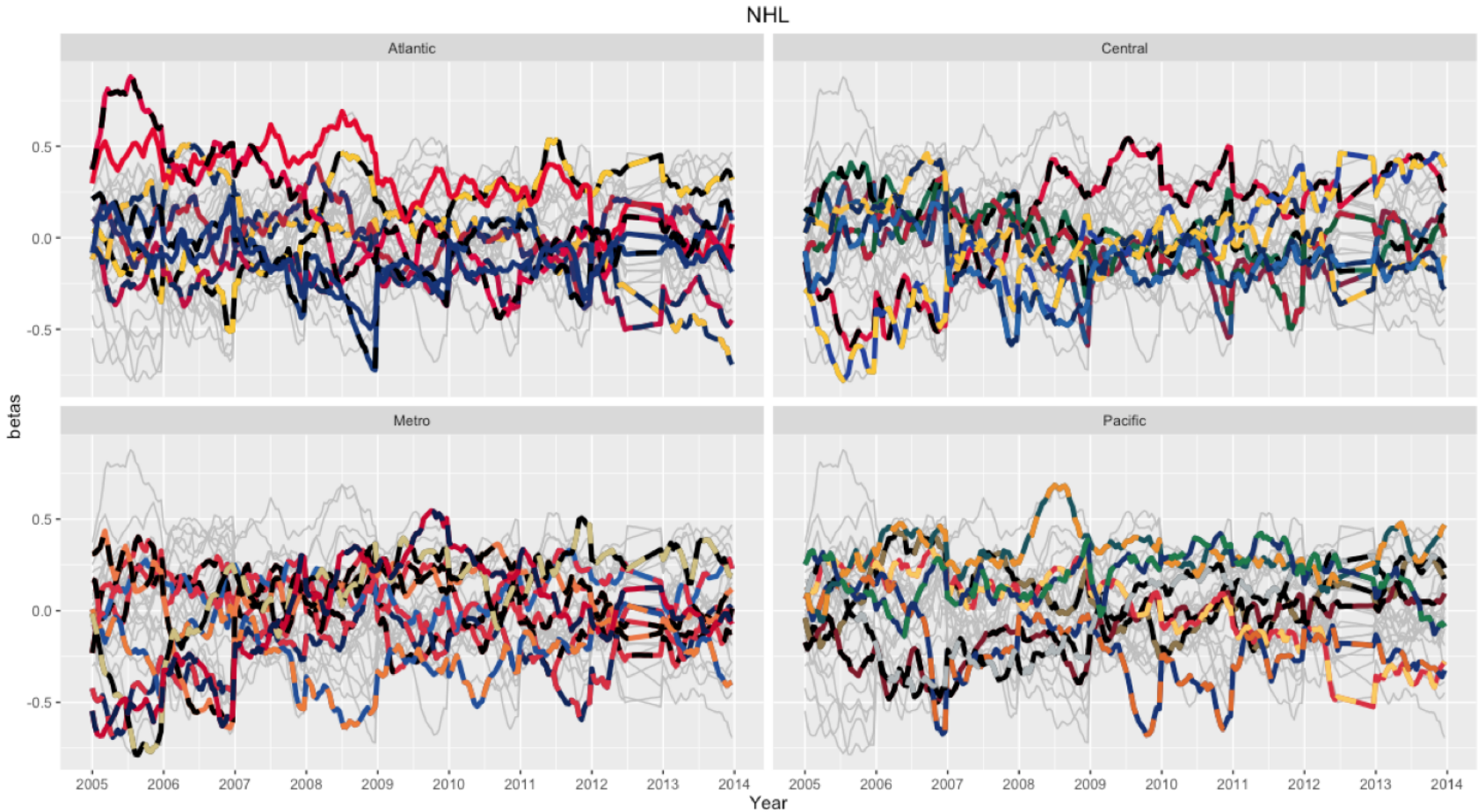


Figure 3: NHL team strength over time

Step 4: Share

- R/RStudio work well with Github repositories
- Easy output to HTML's or PDF's
- Slides for presentations also possible
- Weakness: currently incompatable with Wordpress, other blogging hosts

Final thoughts

1. R or Python are both excellent! You should learn one of them
2. Reproducibility and visualization made easy R
3. Learning new software takes time, but...
4. More hockey work should be reproducible

Extras

Questions???