Stat645

Nullabor

Hadley Wickham

1. Project 3

2. Nullabor package

3. Line up

4. Rorschach

Project 3

Review article

By writing your annotated bibliography, you should have gained an understanding of the key papers in your topic area.

The aim of your review article is make it easy for someone else to gain that understand without reading all the papers.

Nullabor

```
install.packages("nullabor")
library(nullabor)
library(ggplot2)

# Most important methods are
#   * lineup()
#   * rorschach()
#
# Works best in conjunction with ggplot2
```

```r
qplot(mpg, wt, data = mtcars)

# Replace original data
qplot(mpg, wt, data = mtcars) %+%
  lineup(null_permute("mpg"), mtcars)

# Display different samples
qplot(mpg, wt, data = mtcars) %+%
  lineup(null_permute("mpg"), mtcars) +
  facet_wrap(~ .sample)

# Or with ggplot2
inf <- lineup(null_permute("mpg"), mtcars)
ggplot(inf, aes(mpg, wt)) +
  geom_point() +
  facet_wrap(~ .sample)
```

Rorschach

```
# nullabor provides a rorschach method but it's not
# that useful unless you really do want to show some
# real data occasionally.  (Plus rorschach is really
# hard to spell)
#
# Usually easier to generate random data yourself
# with rdply()

rdply(5, data.frame(x = runif(5)))
```

Your turn

Use the Rorschach to explore the distribution of uniform random numbers.

What does a histogram of 100 numbers with a binwidth of 0.1 look like? Is it more or less variable than you expect?

Explore larger numbers and different binwidths.

```
unif <- rdply(10, data.frame(x = runif(100)))
qplot(x, data = unif, binwidth = 0.1) +
    facet_wrap(~ .n)
```

Lineup

```r
threept <- subset(lal, type == "3pt" &
  !is.na(x) & !is.na(y))
threept <- threept[c(".id", "period", "time",
  "team", "etype", "player", "points", "result",
  "x", "y")]

threept <- mutate(threept,
  x = x + runif(length(x), -0.5, 0.5),
  y = y + runif(length(y), -0.5, 0.5),
  r = sqrt((x - 25) ^ 2 + y ^ 2),
  angle = atan2(y, x - 25) * 180 / pi)

# Focus in on shots in the typical range
threept <- subset(threept, r > 20 & r < 39)
```

```
qplot(x, y, data = threept) + coord_equal()

qplot(x, y, data = threept) %+%
   lineup(null_permute("y"), threept) +
   facet_wrap(~ .sample) +
   coord_equal()

# Incredibly obvious!
```

```
qplot(x, y, data = threept) + coord_equal()

qplot(x, y, data = threept) %+%
   lineup(null_permute("y"), threept) +
   facet_wr
   coord_eq

# Incredibly obvious!
```

Method to generate null datasets

```
qplot(x, y, data = threept) + coord_equal()

qplot(x, y, data = threept) %+%
    lineup(null_permute("y"), threept) +
    facet_wr
    coord_e

# Incredibly obvious!
```

Method to generate null datasets

True data

```
qplot(angle, r, data = threept)

# Obviously not independent, but is the underlying
# model quadratic?

qplot(angle, r, data = threept) %+%
  lineup(null_lm(r ~ poly(angle, 2)), threept) +
  facet_wrap(~ .sample)

# Would you reject the null hypothesis that the
# relationship between radius and angle is
# quadratic?
```

Your turn

Experiment with variations of the plot that deal with the overplotting problem. Do you change your mind?

To make things easier, I'd start by generating a smaller set of nulls to practice with. Use the `n` argument to lineup.

```
samps <- lineup(null_lm(r ~ poly(angle, 2)),
  threept, n = 5)

ggplot(samps, aes(angle, r)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~ .sample)

ggplot(samps, aes(angle, r)) +
  geom_boxplot(aes(group = round_any(angle, 15))) +
  facet_wrap(~ .sample)
```

```
samp_sum <- ddply(samps,
  c(".sample", angle = "round_any(angle, 15)"),
  summarise,
  angle = mean(range(angle)),
  lower = quantile(r, 0.25),
  upper = quantile(r, 0.75),
  sd = sd(r)
)

ggplot(samp_sum, aes(angle)) +
  geom_ribbon(aes(ymin = lower, ymax = upper)) +
  facet_wrap(~ .sample)

ggplot(samp_sum, aes(angle, sd)) +
  geom_line() +
  facet_wrap(~ .sample)
```

Your turn

Does position on the court influence the chance of success?

Figure out a good way of displaying this, and then test with visual inference.

```
threept$result <- factor(threept$result)
qplot(x, y, data = threept, colour = result)


qplot(angle, data = threept, fill = result, position
= "fill")
qplot(r, data = threept, fill = result, position =
"fill")


qplot(r, data = threept, fill = result,
  position = "fill", binwidth = 1)


qplot(r, fill = result,
  data = lineup(null_permute("result"), threept),
  position = "fill", binwidth = 1) +
  facet_wrap(~ .sample)
```

Your turn

Download the source code for nullabor from CRAN. How do `null_permute` and `null_lm` work?