Literary Data: Some Approaches

Andrew Goldstone
http://www.rci.rutgers.edu/~ag978/litdata

March 26, 2015. Visualization (1).

txl <- read.csv("three-percent.csv", as.is=T,
 encoding="UTF-8")

txl <- txl %>%
  filter(Year < 2015) %>%
  rename(Language=Lanuage)

txl <- txl %>%
  mutate(
    author_name=str_c(AuthorLN, AuthorFN,
      sep=" ", "",
    ),
    trans_name=str_c(TranslatorLN, TranslatorFN,
      sep=" ", "")
  )

- Derive a data frame of publishers with columns for numbers of authors and numbers of titles
authors_titles <- txl %>%
  group_by(Publisher) %>%
  summarize(authors = n_distinct(author_name),
            titles = n())

Figure 1: Titles vs. authors published for each publisher
**gggrammar**

1. aesthetic mapping
2. statistical transformation (optional)
3. geometric object
4. scale transformation
5. coordinate system
6. guides

---

**aesthetic mapping**

```r
# more quote-free syntax (Wiiiiickham!!!)
p <- ggplot(frm, aes(x=col1, y=col2, color=col3, ...))
```

- `p` abstract plot object
- `x=col1` intent to map the data in `frm$col1` to the aesthetic “x position in the plane” (but does not yet say how)

```r
p <- ggplot(authors_titles, aes(x=titles, y=authors))
```
stat

- **identity**: do nothing
- **bin**: tally up values that occur in a series of ranges
- **smooth**: calculate an approximation (line or curve fit)
- ... (there are many others)

```
p <- p + stat_identity()
```

geom

```
p <- p + geom_point()
```

- **geom_line**: lines connecting observations
- **geom_bar**: rectangular bars
- **geom_boxplot**: box and whiskers
  - (overall shape of each data group)
- **geom_text**: words
  - ... (many others; see Wickham, 56, or docs.ggplot2.org or www.cookbook-r.com/Graphs)
scale

- turn values into graphics instructions
- how many pixels correspond to the difference between 1 title and 2 titles?
- where do we start and end the x axis?

```r
# all taken care of by default values, fortunately
p_default <- p + scale_x_continuous() +
    scale_y_continuous()
```

coordinates

- combine scaled variable mappings into a single plane

```r
p <- p + coord_cartesian()
```
guides

- axes, legends, titles
- `guide_legend` exists, but no one ever uses it
- axis guides are adjusted by `scale_x_continuous`

```r
p <- p + scale_x_continuous(name="titles published") +
    scale_y_continuous(name="authors published")
```

draw something already

```r
print(p)
```

![Figure 2: Our first scatterplot](image)
defaults to the rescue

```r
p <- ggplot(authors_titles, aes(x=titles, y=authors))
p + geom_point()
```

Figure 3: The same, more concisely

- geoms come with assumptions about stats and coords (and position adjustments)
- stats come with assumptions about geoms
- `ggplot` assumes scales based on data types (numeric means continuous)
- evaluating a plot object causes it to be printed
- default guides are meant to be nice on the screen
- infinite tweaking possible through scale parameters and `theme` function
```r
p <- ggplot(authors_titles, aes(x=titles, y=authors)) + 
  geom_point() +
  scale_x_continuous(name="titles published") +
  scale_y_continuous(name="authors published")
```

Figure 4: Some better labels

more than one geom

```r
p + geom_smooth(method="lm")
```

Figure 5: The same, with a regression line
discrete scales

```r
langs <- txl %>% group_by(Language) %>%
  summarize(titles=n()) %>%
  top_n(10, titles)

ggplot(langs, aes(x=Language, y=titles)) + geom_point()
```

Figure 6: An example of a categorical variable. Note y range

```r
# have to set stat and geom at once
ggplot(langs, aes(x=Language, y=titles)) +
  geom_bar(stat="identity")
```

Figure 7: The same, but it’s better with bars
**Figure 8:** Bars bin by default. Too many categories

**yes, you could make a pie**

ggplot(langs, aes(x="", y=titles, fill=Language)) +
geom_bar(width=1, stat="identity") +
coord_polar(theta="y")

**Figure 9:** Ugh
walk the line

titles_year <- txl %>%
group_by(Year) %>%
summarize(titles=n())

ggplot(titles_year, aes(x=Year, y=titles, group=1)) +
geom_line()

Figure 10: Titles per year

doing lines

sp_year <- txl %>%
filter(Language == "Spanish") %>%
group_by(Year, Country) %>%
summarize(titles=n())

You can add both constant and variable aesthetic mappings to each geom:

p <- ggplot(sp_year, aes(x=Year, y=titles)) +
geom_point() +
geom_line(aes(group=Country, color=Country))
sp_year %>% group_by(Country) %>%
  filter(any(titles > 10)) %>%
  ggplot(aes(x=Year, y=titles)) +
  geom_line(aes(group=Country, color=Country)) +
  scale_color_brewer(type="qual") +
  theme(legend.position="bottom")

Figure 11: Still messy…

Figure 12: The same, filtered to clean it up
fancier filtering with joins

top_pubs <- txl %>% group_by(Publisher) %>%
  summarize(count=n()) %>%
  top_n(4) %>%
  select(-count)

Could say:

pub_genre_year <- txl %>%
  filter(Publisher %in% top_pubs$Publisher) %>%
  ...

But instead:

pub_genre_year <- txl %>%
  inner_join(top_pubs) %>%
  group_by(Publisher, Genre, Year) %>%
  summarize(count=n())

p <- ggplot(pub_genre_year,
  aes(x=Year, y=count, group=1)) +
  geom_line() +
  facet_grid(Genre ~ Publisher)

huh??

# a grid of plots, one for each combination
# of x_col and y_col
facet_grid(y_col ~ x_col)

  y ~ x a formula expression, read "y depends on x"
Figure 13: Small multiples