Prediction 2 - tidyverse

Anna Yorozuya

University of Tokyo

June 16, 2022

Table of Contents

- broom package
- modelr package
- tidyr package
- pivot_longer() and pivot_wider()
- visualizing regression
- Today's in-class assignment: conditional-cash-transfer

broom package

what is broom?

- a package in tidymodels package
- converting outputs of baseR functions into tidy data
- for more information, see here.

useful functions

- tidy(): summarizes information about model components
- glance(): reports information about the entire model
- augment(): adds informations about observations to a dataset

useful functions, when used for lm() outputs

- tidy(): returns a data frame in which each row is a coefficient
- glance(): returns a one-row dataframe summary of the model
- augment(): returns the original data with fitted values, residuals, and other observation level stats from the model appended to it.

broom package: example

```
fit <- lm(diff.share ~ d.comp, data = face)
glance(fit)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic     p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>       <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.187         0.180 0.266      27.0 0.000000885     1  -10.5  27.0  35.3
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

broom package: example

```
tidy(fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic     p.value
##   <chr>          <dbl>     <dbl>     <dbl>       <dbl>
## 1 (Intercept)   -0.312    0.0660     -4.73 0.00000624
## 2 d.comp         0.660    0.127       5.19 0.000000885
```

broom package: example

```
augment(fit) %>% head()
```

```
## # A tibble: 6 x 8
##   diff.share d.comp .fitted  .resid     .hat .sigma  .cooksd .std.resid
##        <dbl>  <dbl>   <dbl>   <dbl>    <dbl>  <dbl>    <dbl>      <dbl>
## 1     0.210   0.565  0.0606  0.150  0.00996   0.267 0.00160       0.564
## 2     0.119   0.342 -0.0864  0.206  0.0129    0.267 0.00394       0.778
## 3     0.0499  0.612  0.0922 -0.0423 0.0123    0.268 0.000158     -0.160
## 4     0.197   0.542  0.0454  0.151  0.00922   0.267 0.00151       0.570
## 5     0.496   0.680  0.137   0.359  0.0174    0.266 0.0163        1.36
## 6    -0.350   0.321 -0.101  -0.249  0.0143    0.267 0.00644      -0.941
```

modelr package

what is modelr?

- a package for helping modeling in tidyverse framework, especially with pipes
- for more information, see here

useful functions

- add_predictions(): add the predictions to the original data
- add_residuals(): add the residuals to the original data
- data_grid(): create a data set containing every unique combination of the specified columns from the old data set.
- spread_predictions(): generate two sets of predictions for a new tibble of data

modelr package: example

```
fit2 <- lm(Buchanan00 ~ Perot96, data = florida)
florida_fit2 <- florida %>%
  add_predictions(fit2) %>%
  add_residuals(fit2)
head(florida_fit2)
```

```
##     county Clinton96 Dole96 Perot96 Bush00 Gore00 Buchanan00       pred
## 1  Alachua     40144  25303    8072  34124  47365        263  291.25196
## 2    Baker      2273   3684     667   5610   2392         73   25.30108
## 3      Bay     17020  28290    5922  38637  18850        248  214.03462
## 4 Bradford      3356   4038     819   5414   3075         65   30.76017
## 5  Brevard     80416  87980   25249 115185  97318        570  908.16461
## 6  Broward    320736 142834   38964 177323 386561        788 1400.73939
##        resid
## 1  -28.25196
## 2   47.69892
## 3   33.96538
## 4   34.23983
## 5 -338.16461
## 6 -612.73939
```

modelr package: example

```r
fit <- lm(primary2006 ~ messages, data = social)
unique_messages <- data_grid(social, messages) %>%
  add_predictions(fit)
unique_messages
```

```
## # A tibble: 4 x 2
##   messages    pred
##   <chr>      <dbl>
## 1 Civic Duty 0.315
## 2 Control    0.297
## 3 Hawthorne  0.322
## 4 Neighbors  0.378
```

tidyr package

what is tidyr?

- a package in tidyverse helping to tidy data
- for more data, see here

useful function

- crossing(): produce a new data set with all combinations of the specified variable values

tidyr package: example

```
fit.age <- lm(primary2006 ~ age * messages, data = social.neighbor)
ate.age <- tidyr::crossing(age = seq(from = 20, to = 80, by = 20),
         messages = c("Neighbors", "Control")) %>%
  add_predictions(fit.age) %>%
  pivot_wider(names_from = messages,
              values_from = pred) %>%
  mutate(diff = Neighbors - Control)
ate.age
```

```
## # A tibble: 4 x 4
##     age Control Neighbors   diff
##   <dbl>   <dbl>     <dbl>  <dbl>
## 1    20   0.169     0.231 0.0611
## 2    40   0.249     0.323 0.0737
## 3    60   0.329     0.416 0.0863
## 4    80   0.409     0.508 0.0988
```

pivot_longer() and pivot_wider()

pivot_longer()

- increase the number of rows, while decreasing the number of columns
- argument cols = x: specify the columns (x) to pivot into longer formats
- argument names_to: name the new columns for storing data from the columns specified in the cols argument.

pivot_wider()

- increase the number of columns, while decreasing the number of rows
- argument names_from: describe which column to get the name of the output column.

pivot_longer() and pivot_wider(): example

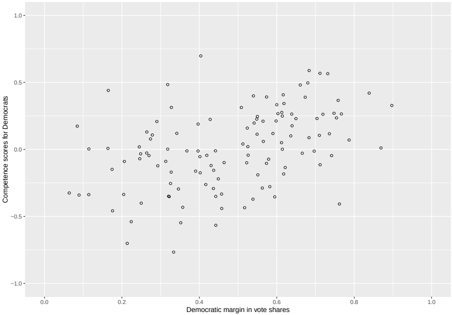
```
women %>%
  group_by(reserved) %>%
  summarize(irrigation = mean(irrigation),
            water = mean(water)) %>%
  pivot_longer(names_to = "variable", - reserved) %>%
  pivot_wider(names_from = reserved) %>%
  rename("not_reserved" = `0`,
         "reserved" = `1` ) %>%
  mutate(diff = reserved - not_reserved)
```

```
## # A tibble: 2 x 4
##   variable   not_reserved reserved   diff
##   <chr>             <dbl>    <dbl>  <dbl>
## 1 irrigation         3.39     3.02 -0.369
## 2 water             14.7     24.0   9.25
```

visualizing regression 1: geom_point() + geom_abline()

```
ggplot() +
  geom_point(data = face,
             mapping = aes(x = d.comp, y = diff.share), shape = 1) +
  geom_abline(slope = coef(fit)["d.comp"],
              intercept = coef(fit)["(Intercept)"]) +
  scale_y_continuous("Competence scores for Democrats",
                     breaks = seq(-1, 1, by = 0.5), limits = c(-1, 1)) +
  scale_x_continuous("Democratic margin in vote shares",
                     breaks = seq(0, 1, by = 0.2), limits = c(0, 1))
```

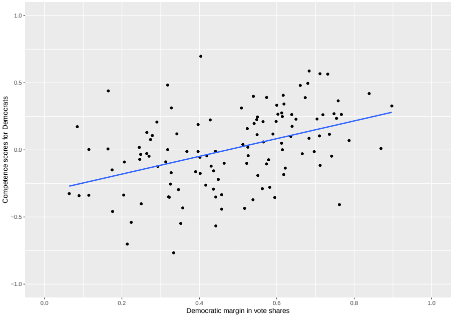
```
## Warning: Removed 1 rows containing missing values (geom_abline).
```



visualizing regression 2: geom_point() + geom_smooth()

```
ggplot(data = face, mapping = aes(x = d.comp, y = diff.share)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_y_continuous("Competence scores for Democrats",
                     breaks = seq(-1, 1, by = 0.5), limits = c(-1, 1)) +
  scale_x_continuous("Democratic margin in vote shares",
                     breaks = seq(0, 1, by = 0.2), limits = c(0, 1))
```

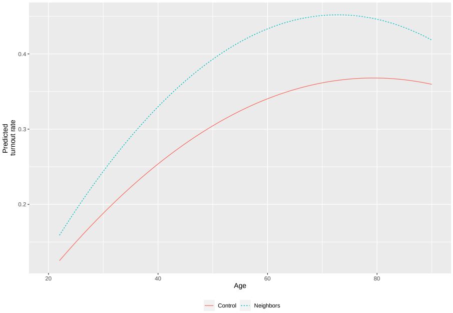
```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



visualizing regression 3: `geom_point() + geom_line()`

```r
ggplot(y.hat, aes(x = age, y = pred)) +
  geom_line(aes(linetype = messages,
                color = messages)) +
  labs(color = "",
       linetype = "", y = "Predicted \nturnout rate",
       x = "Age") +
  xlim(20, 90) +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 34 row(s) containing missing values (geom_path).
```



References

- “Quantitative Social Science: An Introduction” - Kosuke Imai
- “Quantitative Social Science: An Introduction in Tidyverse” - Kosuke Imai and Nora Webb Williams
- R for data science - H.Wickham and G.Grolemund
- broom package
- modelr package
- tidyr package