Regression and Other Stories: Continuous Predictor

Regression Example

We will once again use the Brazilian beer dataset to illustrate the regression process.

```
beer <- read_csv('http://math.montana.edu/ahoegh/Data/Brazil_cerveja.csv')
```

- Fit Model

```
stan_fit <- stan_glm(consumed ~ max_tmp, data = beer, refresh = 0)
print(stan_fit)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      consumed ~ max_tmp
##  observations: 365
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 7.9    1.1
## max_tmp     0.7    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 3.4    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

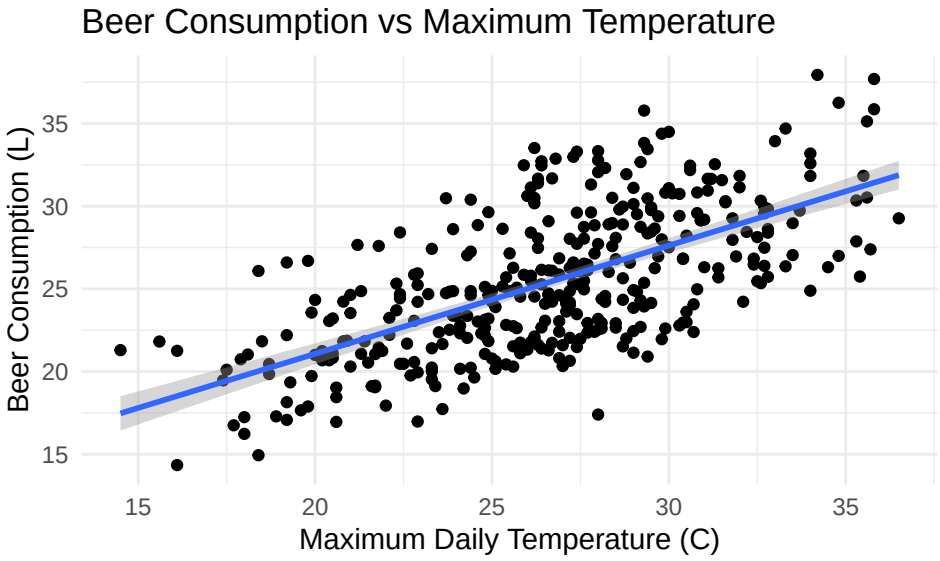
- the fitted regression line is

- At $x = 0$

- Each additional degree (maximum temperature) corresponds

- The standard errors around the coefficients are quite small.

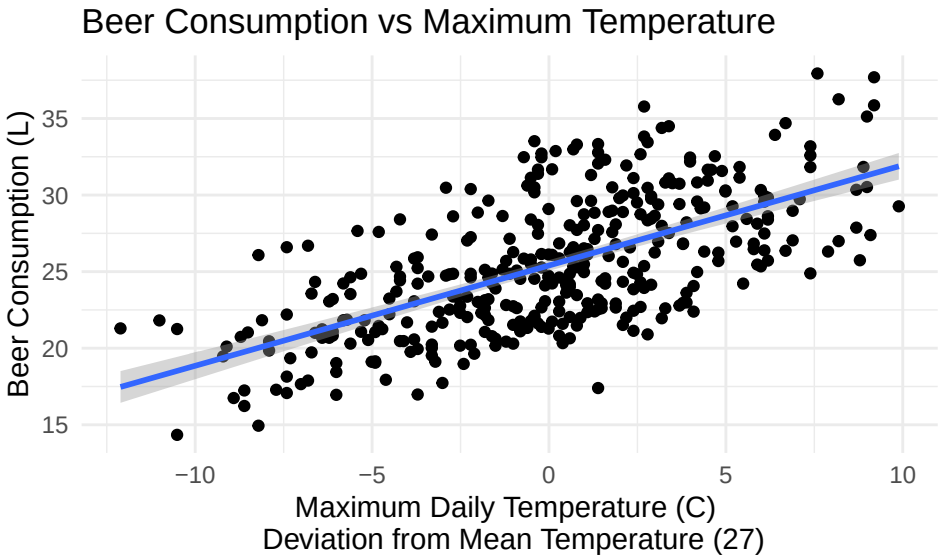
- The estimated *residual standard deviation* is 3.4.




```
ave_tmp <- beer %>% summarize(ave_tmp = mean(max_tmp)) %>% pull()
```

Centering Data We can center the temperature by subtracting the average temperature. Thus the new variable corresponds to deviation from the average temperature.

```
beer <- beer %>% mutate(tmp_centered = max_tmp - ave_tmp)
```




```
stan_fit <- stan_glm(consumed ~ tmp_centered, data = beer, refresh = 0)
print(stan_fit)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      consumed ~ tmp_centered
##  observations: 365
##  predictors:   2
## ------
##              Median MAD_SD
## (Intercept)  25.4    0.2
## tmp_centered  0.7    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 3.4    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

- the fitted regression line is

- At $x^{'} = 0$

- Each degree different from the daily average maximum temperature corresponds

- The standard errors around the coefficients are quite small.

- The estimated *residual standard deviation* is 3.4. To interpret this value, roughly 68% of the daily consumption values will be within $\pm$ 3.4 liters of the fitted regression line and 95% will fall within $\approx \pm 2 \times 3.4$ liters of the regression line.

Comparisons of a mean and linear models

Consider comparing the mean beer consumption between weekend and weekdays.

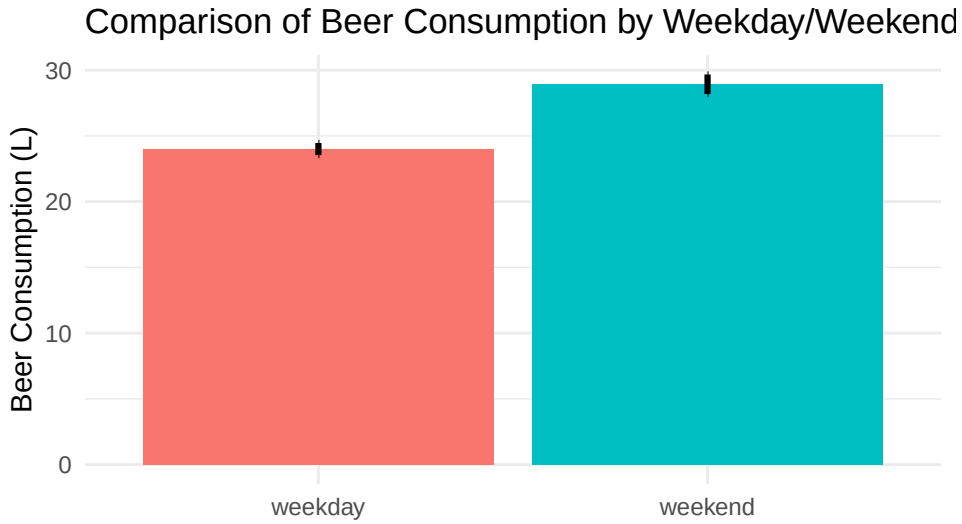


Figure 1: Comparison of beer consumption by day of week.

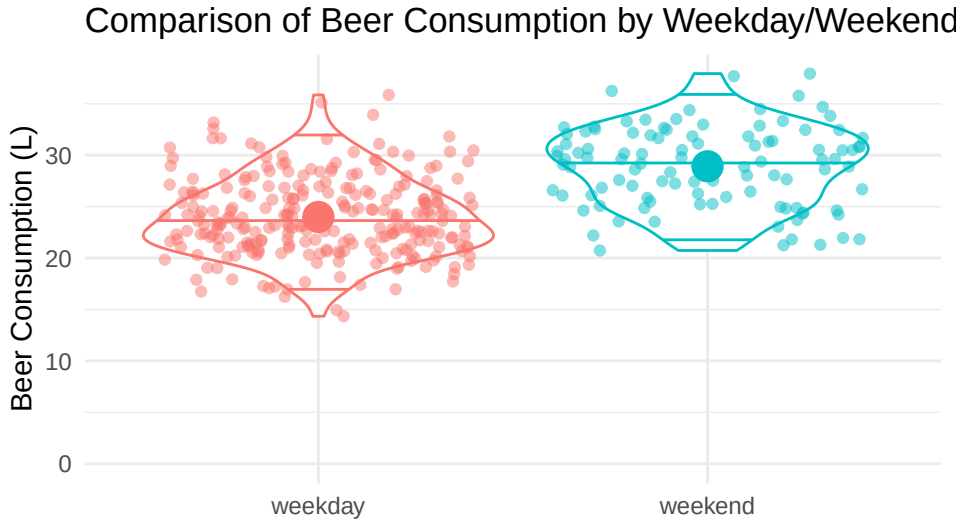


Figure 2: Comparison of beer consumption by day of week. The lines on the violin plots correspond to the median, and the .025 and .975 quantiles. The large circles represent the mean comsumption for each group.

A t-test is a common procedure for comparing whether the mean differs between two populations.

```
weekend <- beer %>% filter(weekend == 1) %>% dplyr::select(consumed) %>% pull()
weekday <- beer %>% filter(weekend == 0) %>% dplyr::select(consumed) %>% pull()
t.test(weekend, weekday)
```

```
##
##  Welch Two Sample t-test
##
## data:  weekend and weekday
## t = 11.123, df = 187.62, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  4.051098 5.797899
## sample estimates:
## mean of x mean of y
##  28.92272  23.99822
```

Formally this can be expressed as a linear model.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where y_i is the consumption on day i, x_i is a indicator variable for whether day i is a weekend.

```
beer %>% mutate(weekend = as.factor(weekend)) %>%
stan_glm(consumed ~ weekend, data = ., refresh = 0)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      consumed ~ weekend
##  observations: 365
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 24.0    0.2
## weekend1     4.9    0.4
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 3.8    0.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```