

# Lecture 10: Gelman Hill Ch 7

## Simulation-Based Model Assessment

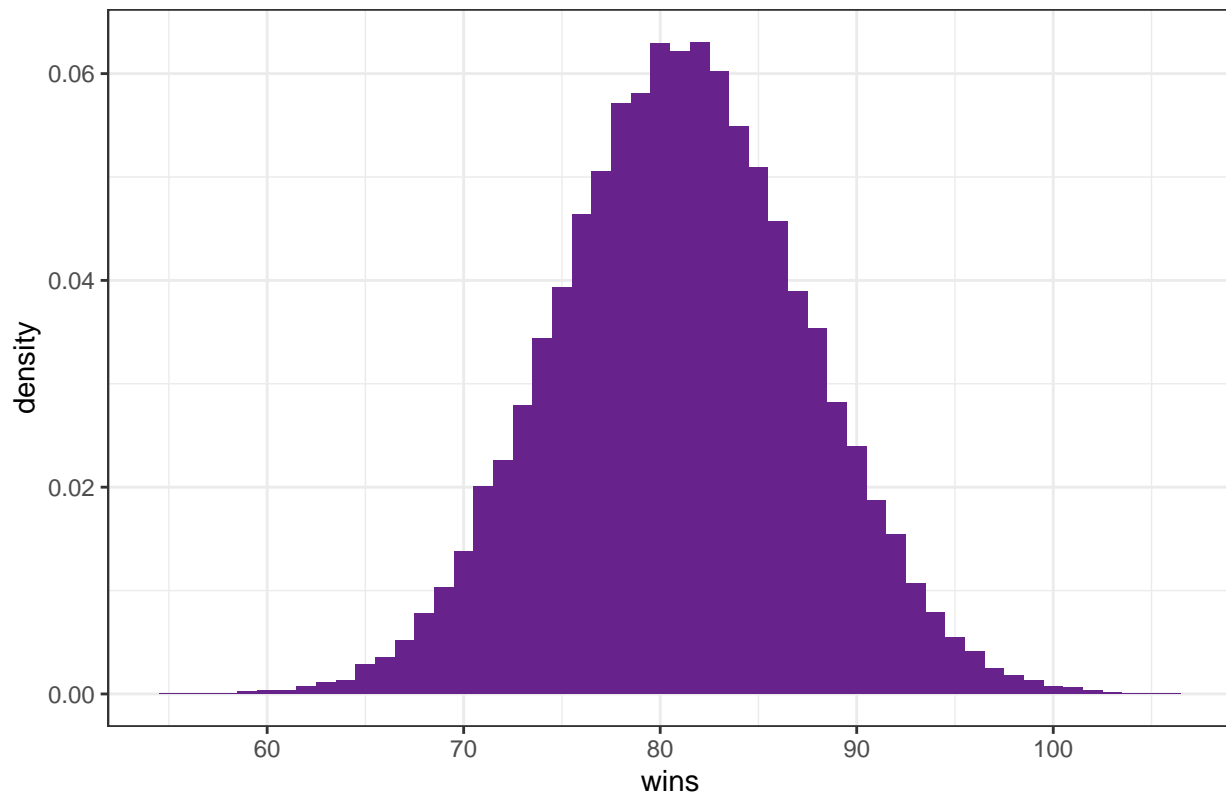
### Simulation of Probabilistic Models

Suppose the Colorado Rockies have a talent level that corresponds to winning 50% of their games. What are the reasonable expectation for the number of games the Rockies will win this season (out of 162).

Specifically, we are interested in the distribution of outcomes that could be generated, given the Rockies are expected to win 50% of their games?

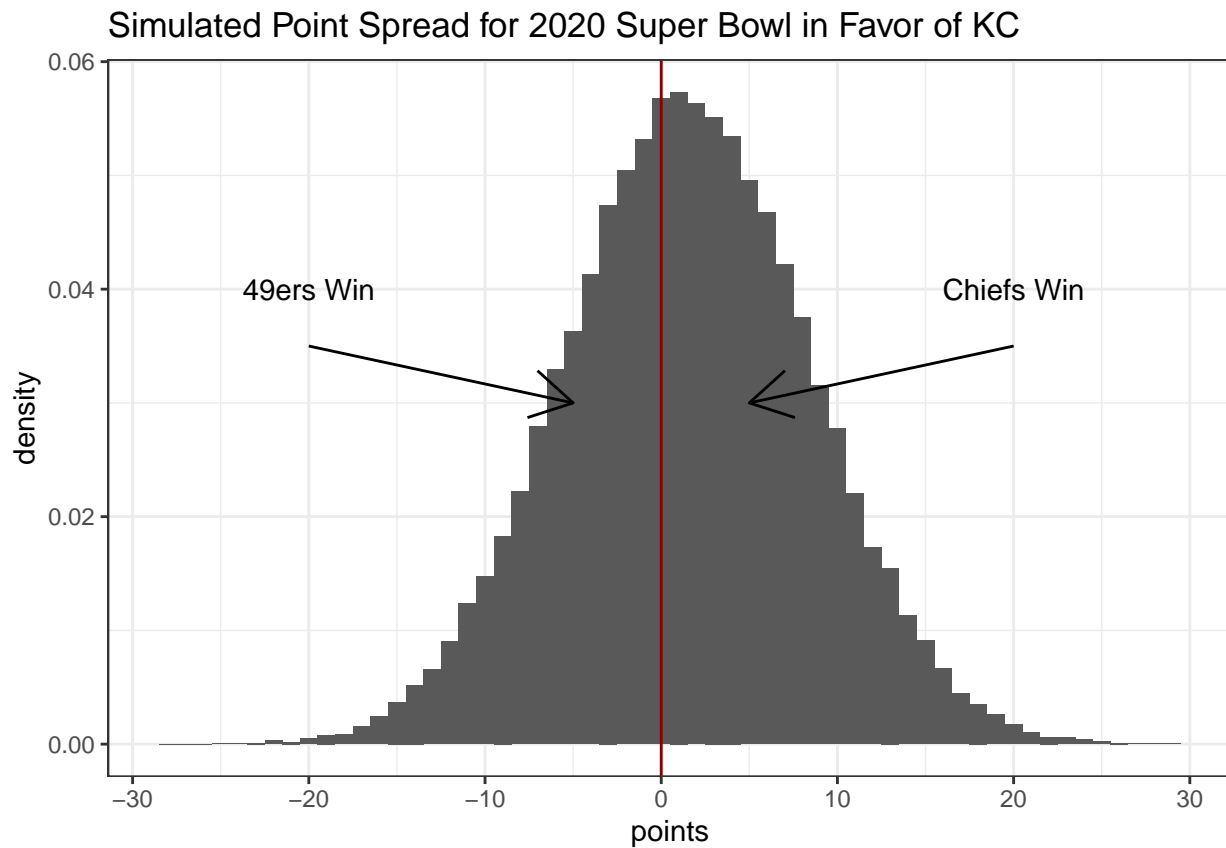
In one scenario, the Rockies could win 82. However, we are interested in the distribution of possible outcomes.

Distribution of Wins for Colorado Rockies with 50% win probability



We can do the same thing for a continuous variable. Consider the 2020 Super Bowl where the Kansas City Chiefs were 1.5 point favorites. Assume that this point differential can be translated to a normal distribution with standard deviation of 7.

Then the range of possible outcomes can be simulated...



This can also be used to calculate the winning probability (under the assumed model), where the Chiefs would win with probability ‘

## Summarizing linear regression with simulation

```
set.seed(02262020)
beer <- read_csv('http://math.montana.edu/ahoegh/Data/Brazil_cerveja.csv') %>%
  mutate(precip = scale(precip), max_tmp = scale(max_tmp))
```

```
lm_beer <- lm(consumed ~ precip + max_tmp + weekend, data = beer)
display(lm_beer)
```

```
## lm(formula = consumed ~ precip + max_tmp + weekend, data = beer)
##               coef.est coef.se
## (Intercept)  23.92      0.14
## precip       -0.71      0.12
## max_tmp       2.89      0.12
## weekend        5.18      0.27
## ---
## n = 365, k = 4
## residual sd = 2.33, R-Squared = 0.72
```

Confidence intervals can be analytically computed... (but still require distributional assumptions or CLT asymptotics)

```
confint(lm_beer)
```

```
##                2.5 %    97.5 %
## (Intercept) 23.6409989 24.2075184
## precip      -0.9539859 -0.4737965
## max_tmp      2.6460541  3.1266325
## weekend       4.6531170  5.7150484
```

Simulation can also be used construct intervals

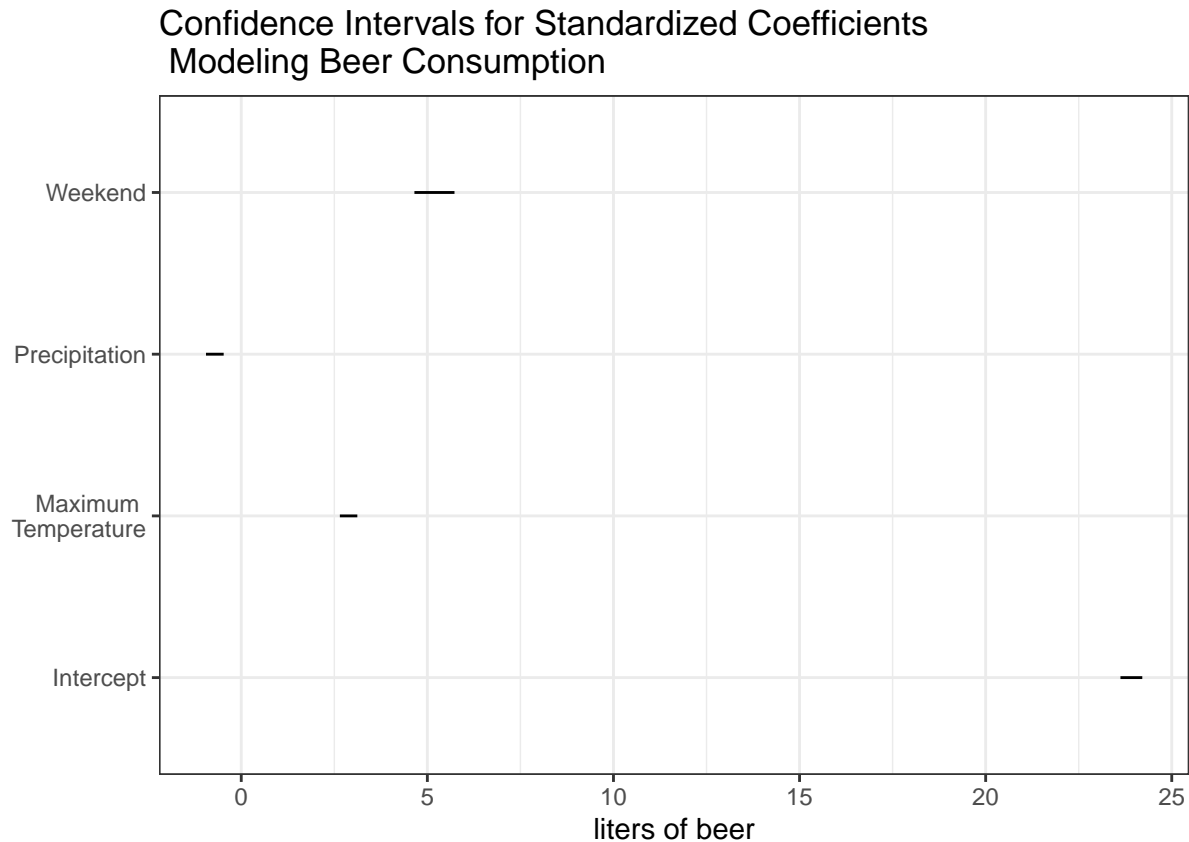
```
n.sims <- 1000
sim_vals <- arm::sim(lm_beer, n.sims)
head(coef(sim_vals))
```

```
##      (Intercept)      precip max_tmp weekend
## [1,]  24.08906 -0.7182283  2.816890 5.146346
## [2,]  24.00334 -0.8814225  2.770705 4.979809
## [3,]  24.11486 -0.5414572  2.900834 5.396089
## [4,]  23.95317 -0.7030344  3.013430 4.869024
## [5,]  24.04089 -0.7942222  2.887384 4.957233
## [6,]  23.93099 -0.7383508  2.973810 5.349378
```

```
ci <- apply(coef(sim_vals), 2, quantile, probs = c(.025, .975)) # simulation from arm package
round(ci, 2)
```

```
##      (Intercept) precip max_tmp weekend
## 2.5%      23.62  -0.95    2.65    4.65
## 97.5%      24.21  -0.47    3.12    5.73
```

```
tibble(var = colnames(coef(sim_vals)), lower = as.numeric(ci[1,]), upper = as.numeric(ci[2,])) %>%
  ggplot() + geom_segment(aes(x=lower, xend = upper, y = var, yend = var)) +
  ggtitle("Confidence Intervals for Standardized Coefficients \n Modeling Beer Consumption") +
  xlab('liters of beer') + theme_bw() + ylab('') +
  scale_y_discrete(labels = c('Intercept', 'Maximum \n Temperature', 'Precipitation', 'Weekend'))
```



Interpret this figure and explain the results to an owner of a liquor store.

As an aside, Megan Higgs has a blog about critical thinking in statistical inference (critical inference). You should read it <https://critical-inference.com/>. Here is an image from a discussion about power, see ([<https://critical-inference.com/sample-size-without-power-yes-its-possible/>]<https://critical-inference.com/sample-size-without-power-yes-its-possible/>)

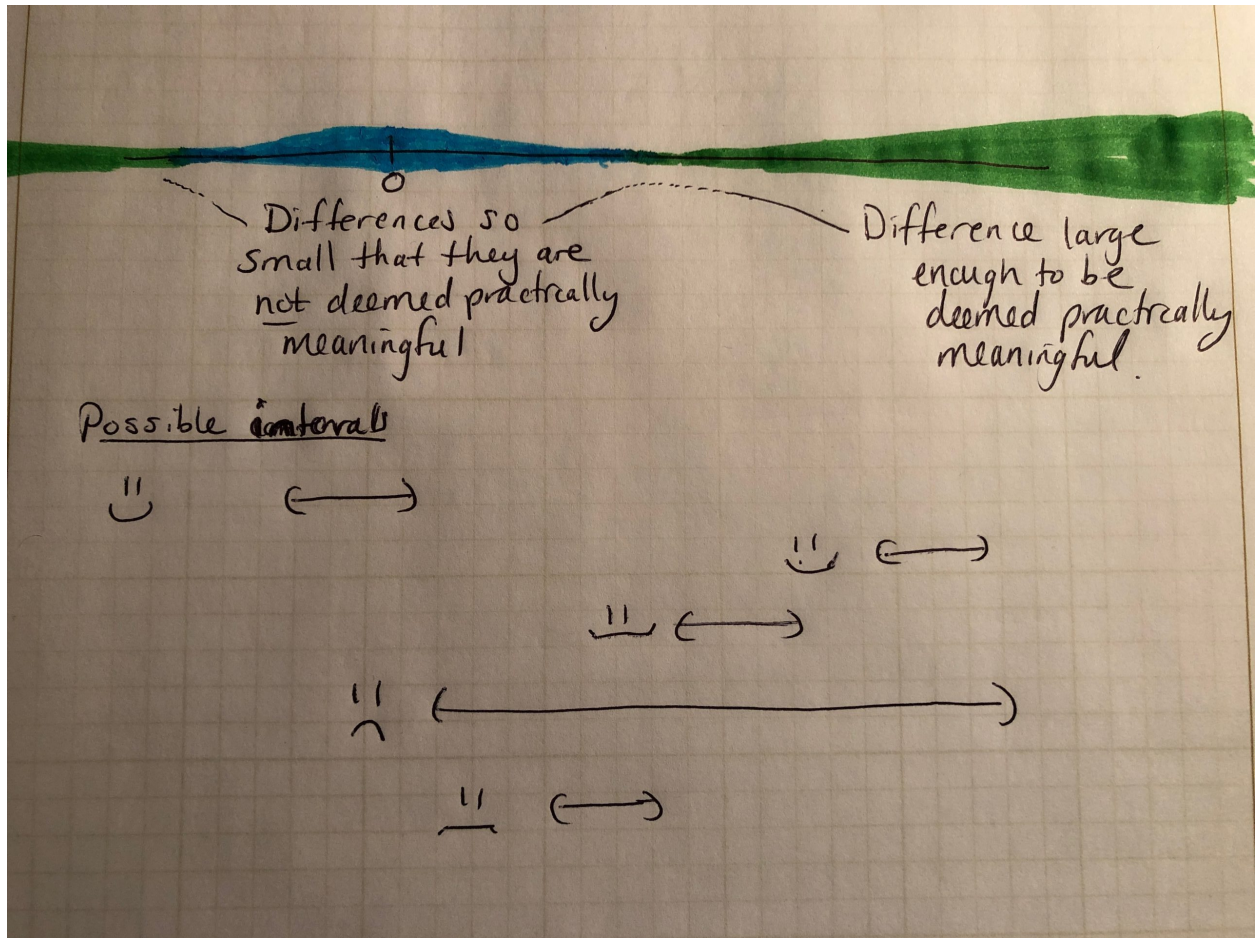


Figure 1: Critical Inference

## Simulation for predictive inference

Suppose the liquor store owner is interested in the following results:

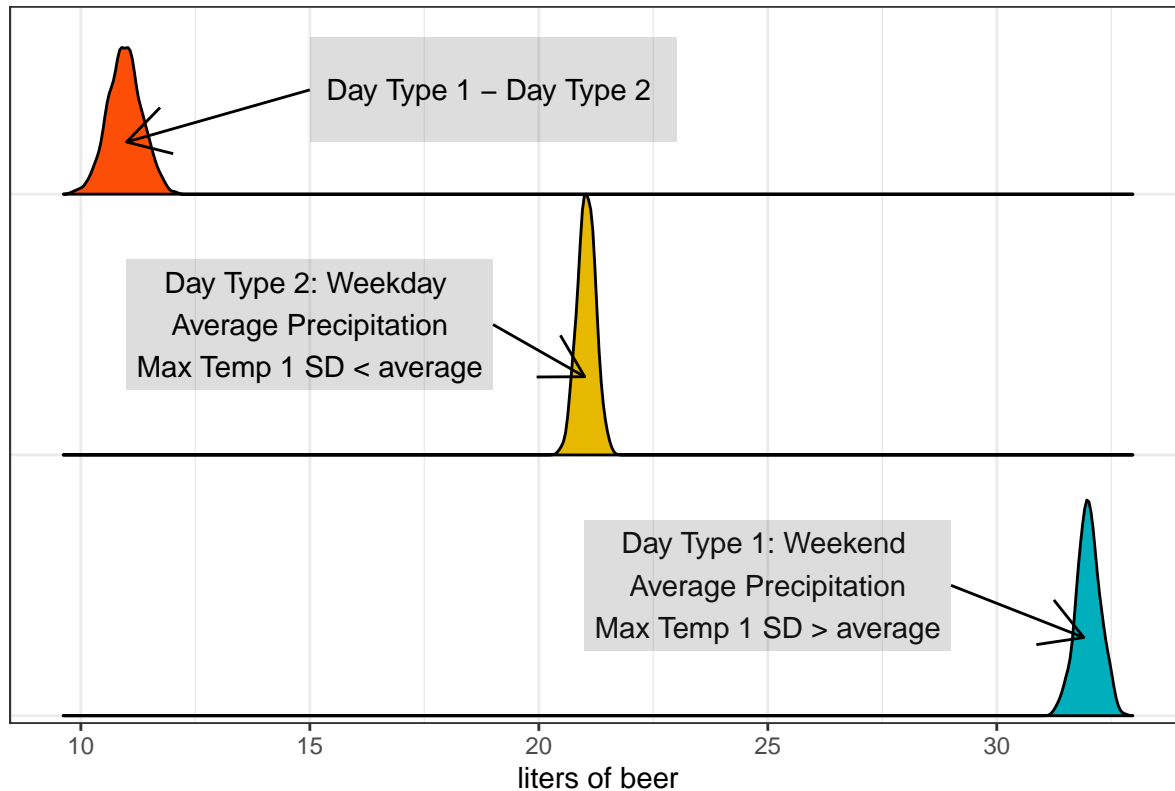
1. the mean predicted consumption on a weekend with average precipitation and temperature 1 sd greater than average.
2. the mean predicted consumption on a week day with average precipitation and temperature 1 sd less than average.
3. the mean predicted difference in consumption between the two days specified in point 1 and point 2.

Answering these questions is easy with simulation...

```
post_plot <- tibble( values = c(pred1, pred2, pred3), type = rep(c('pred1','pred2','pred3'),
                                                                each = n.sims)) %>%
  ggplot(aes(x = values, y = type, fill = type)) + geom_density_ridges2(scale = 1) +
  scale_y_discrete(expand = c(0.01, .01,0.01,.7)) + ggtitle('Average Predicted Beer Consumption in Brazil') +
  ylab('') + xlab('liters of beer') +
  annotate('text',label = 'Day Type 1: Weekend \n Average Precipitation \n Max Temp 1 SD > average ',
          x=25, y = 1.5) +
  annotate('rect',xmin = 21,xmax = 29, ymin = 1.25, ymax = 1.75, alpha = .2) +
  theme_bw() + theme(axis.text.y = element_blank(),axis.ticks.y = element_blank(),
                    legend.position = "none") +
  annotate('segment', x=29, xend = 31.9, y = 1.5, yend=1.3,arrow = arrow()) +
  annotate('rect', xmin = 11,xmax = 19, ymin = 2.25, ymax = 2.75, alpha = .2) +
  annotate('segment', x=19, xend = 21, y = 2.5, yend=2.3,arrow = arrow()) +
  annotate('text',label = 'Day Type 2: Weekday \n Average Precipitation \n Max Temp 1 SD < average ',
          x=15, y = 2.5) +
  annotate('rect', xmin = 15,xmax = 23, ymin = 3.2, ymax = 3.6, alpha = .2) +
  annotate('segment', x=15, xend = 11, y = 3.4, yend=3.2,arrow = arrow()) +
  annotate('text',label = 'Day Type 1 - Day Type 2 ',x=19, y = 3.4) +
  scale_fill_manual(values = c("#00AFBB", "#E7B800", "#FC4E07"))
```

post\_plot

## Average Predicted Beer Consumption in Brazil

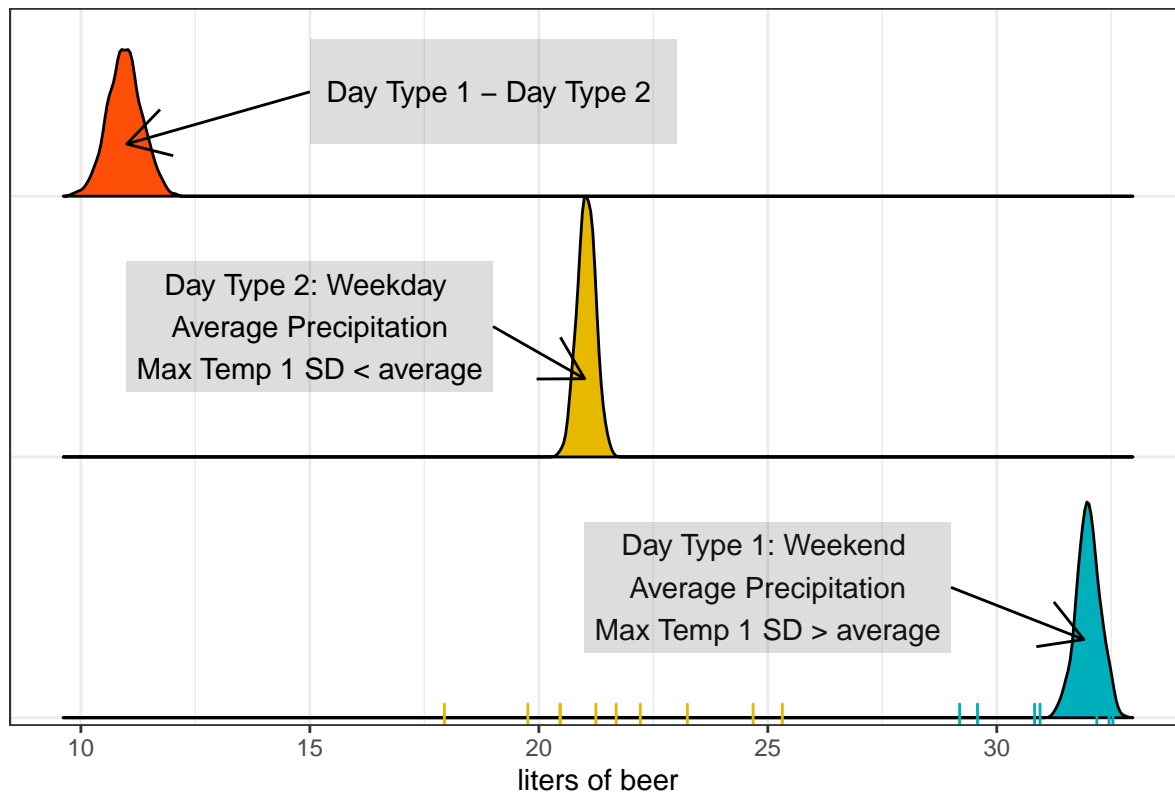


So what exactly is the “mean predicted consumption...”? Does this correspond to predictions for a single day?

Put another way what does the uncertainty in these density plots correspond to?

```
post_plot +  
  geom_rug(inherit.aes = F, data = beer %>% filter(weekend == 1 & max_tmp > .9 & max_tmp < 1.1),  
    aes(x = consumed, y = NULL), sides = 'b', colour = "#00AFBB") +  
  geom_rug(inherit.aes = F, data = beer %>% filter(weekend == 0 & max_tmp < -.9 & max_tmp > -1.1),  
    aes(x = consumed, y = NULL), sides = 'b', colour = "#E7B800")
```

## Average Predicted Beer Consumption in Brazil



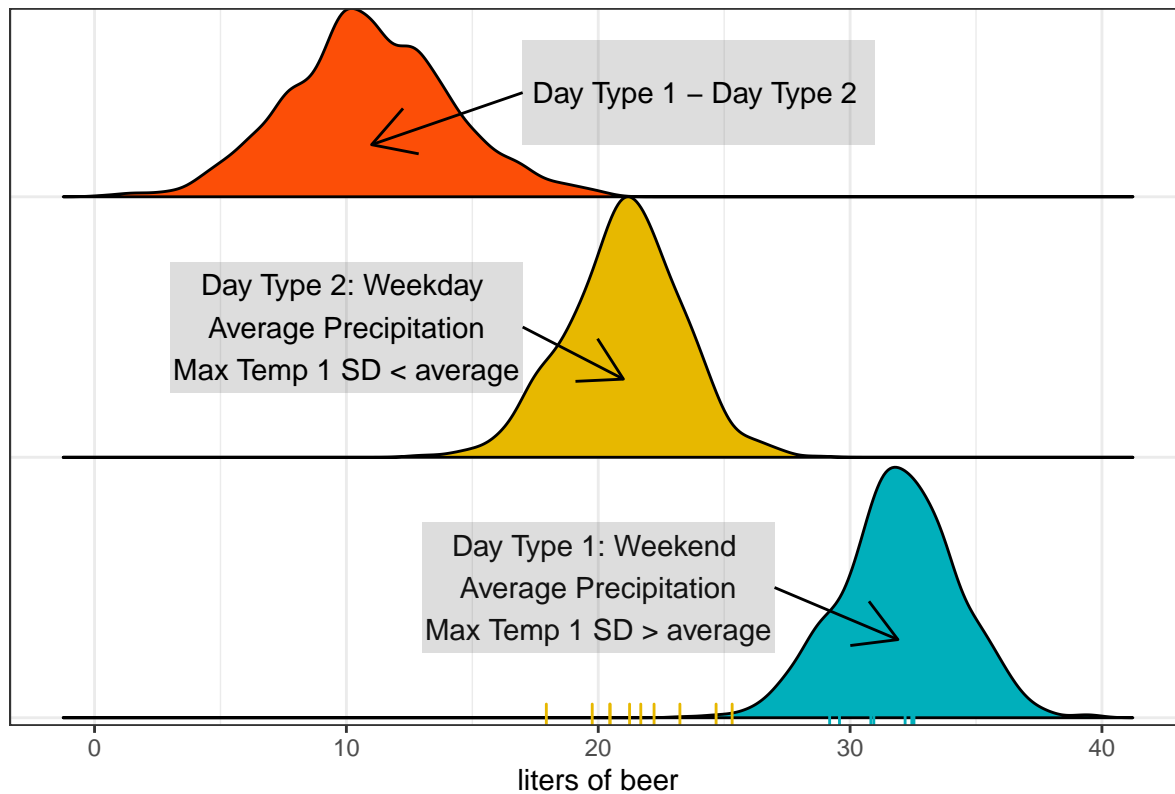
### Prediction for a single data point

Our model for a single data point is

```
pred1_indiv <- rnorm(n = n.sims, mean = pred1, sd = sigma.hat(sim_vals) )  
pred2_indiv <- rnorm(n = n.sims, mean = pred2, sd = sigma.hat(sim_vals) )  
pred3_indiv <- pred1_indiv - pred2_indiv
```



## Daily Predicted Beer Consumption in Brazil



## Simulation Procedure

Simulation can be conducted using the `sim()` function in `arm`

1. Use classical regression of the  $n$  data points with  $k$  predictors to compute the estimate of regression coefficients

2. For each of the `n.sims` iterations, create the coefficient vector  $\hat{\underline{\beta}}^{iter}$  and residual standard deviation  $\hat{\sigma}^{iter}$ .

a.  $\hat{\sigma}^{iter} =$

b.  $\hat{\underline{\beta}}^{iter} \sim$

Note: this is referred to as