Parameter Interpretation - IFLS data

Sarah Teichman

03/05/20

```
fit <- fit_ifls
dat <- deflate_y_dat
post <- as.data.frame(fit)
fit_summ <- as.data.frame(summary(fit)$summary)
```

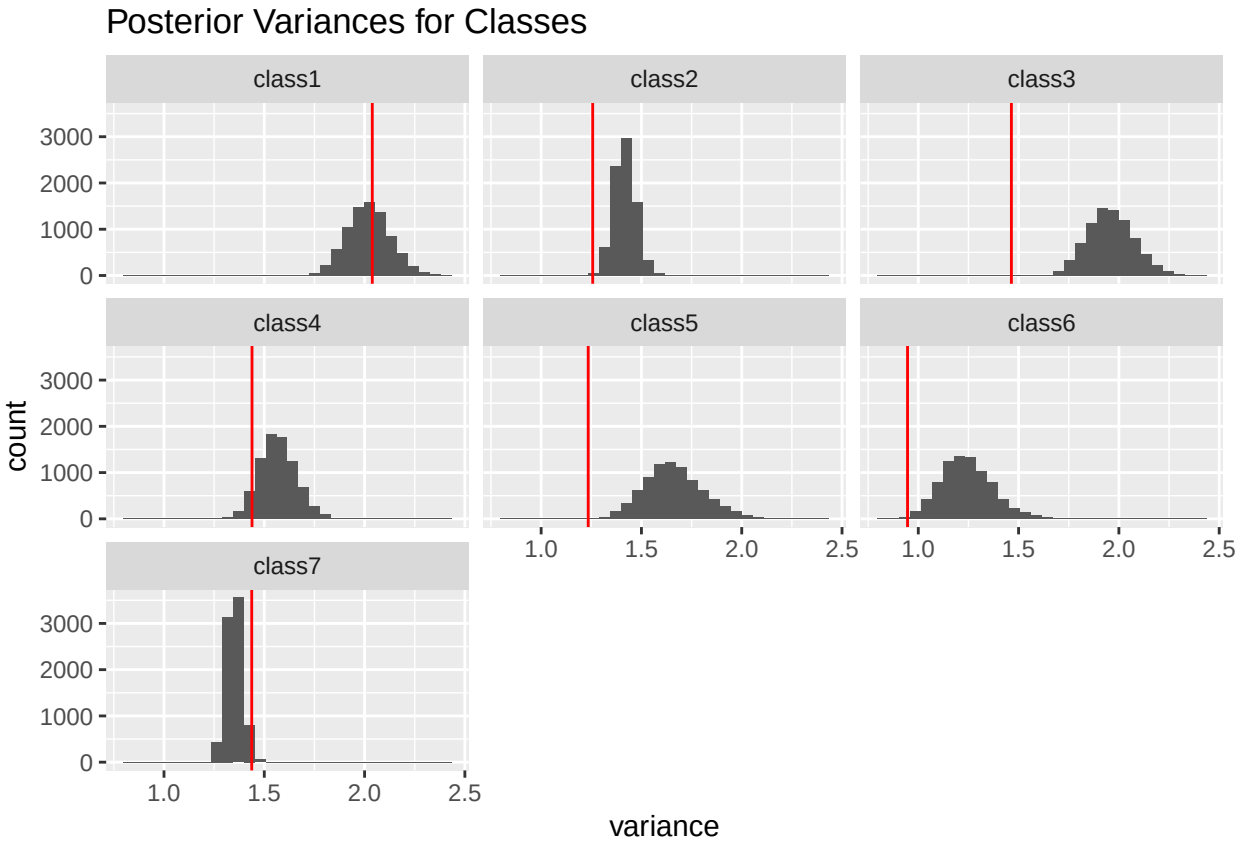
```
K <- 7
Ti <- 3
N <- 1973
```

Posterior histograms for σ_k^2

Plotting posterior histograms from σ_k^2 parameters.

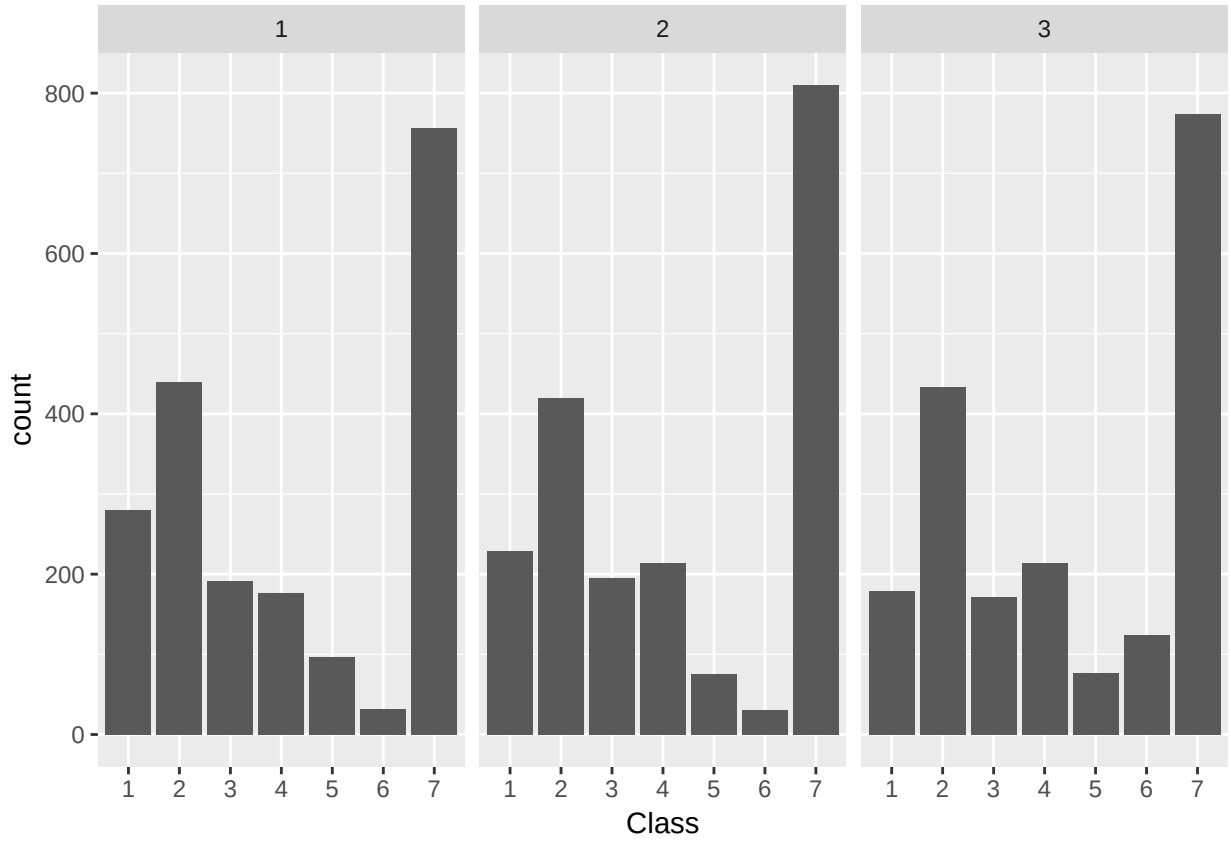
Computing empirical wage variances across firm classes.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



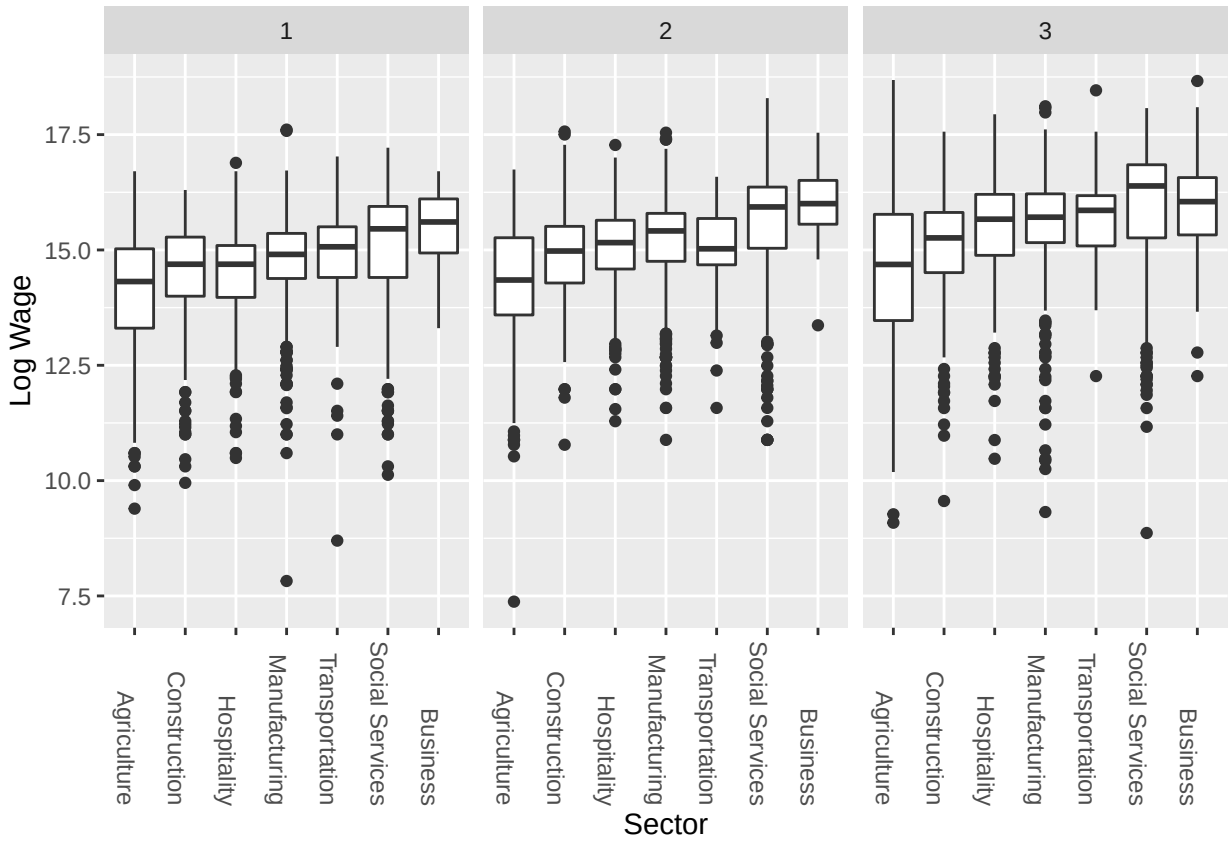
Size of each Class over Time

```
job_long <- as.data.frame(job) %>% pivot_longer(cols = c(V1,V2,V3),names_to = "time",
                                                names_prefix = "V")
ggplot(job_long, aes(x = as.factor(value))) + geom_bar() + facet_wrap(~time) +
  xlab("Class")
```



Log wages by Class and Time

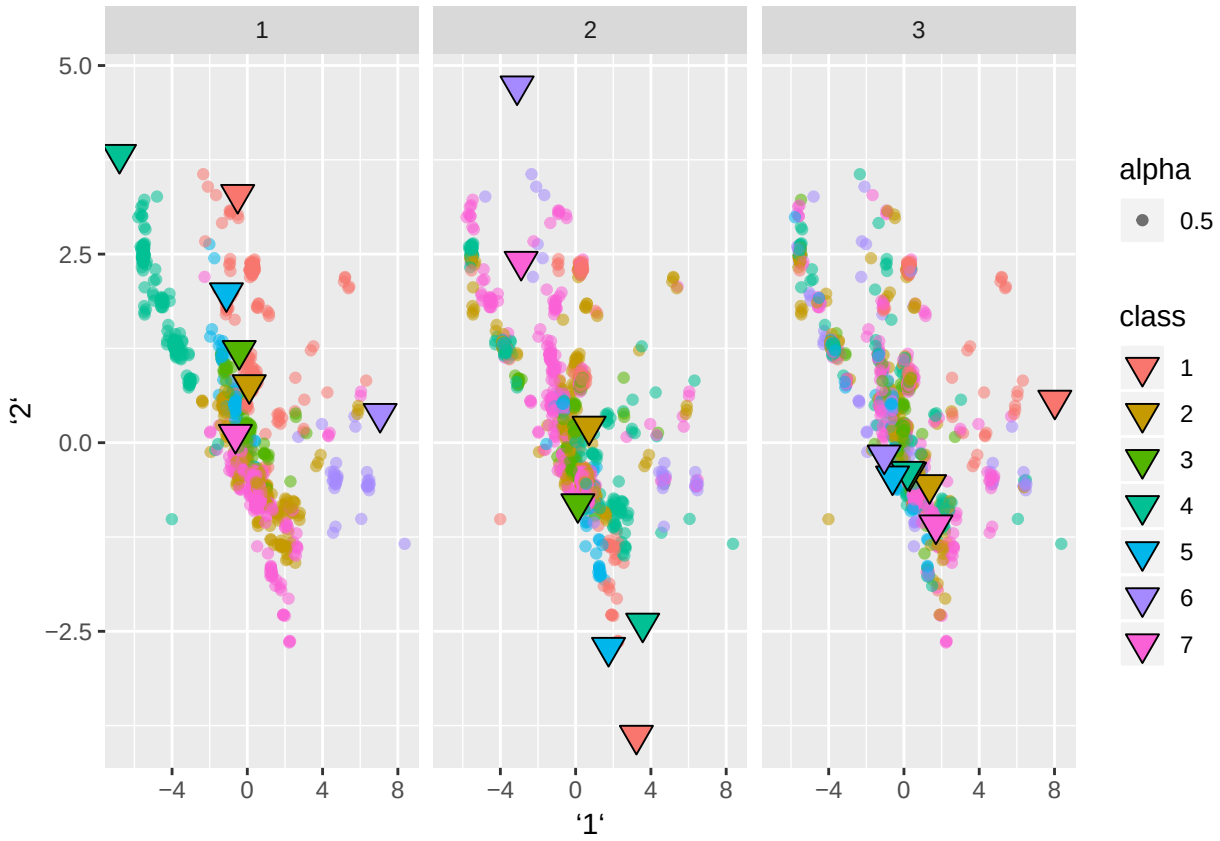
```
labs <- c("Agriculture","Manufacturing","Construction","Hospitality",
          "Transportation","Business","Social Services")
wage_job_dat <- wage_job_dat %>%
  mutate(time_job = as.factor(paste0(time,"-",job)))
meds <- wage_job_dat %>% group_by(time, job) %>%
  summarise(med = median(log_wage)) %>%
  arrange(time, med)
wage_job_dat$job <- factor(wage_job_dat$job, levels = meds$job[1:7])
ggplot(wage_job_dat, aes(x = job, y = log_wage)) + geom_boxplot() +
  scale_x_discrete(labels= labs[meds$job[1:7]]) +
  theme(axis.text.x = element_text(angle = 270, hjust = 1)) +
  facet_wrap(~time) + xlab("Sector") + ylab("Log Wage")
```



Plot latent space positions

```
med_z <- data.frame(par = rownames(fit_summ), med = fit_summ[,6]) %>%
  filter(grepl("z\\[", par)) %>%
  cSplit("par",direction = "wide") %>%
  mutate(worker = stri_extract_first_regex(par_1,"[0-9]+"),
         dim = stri_extract_first_regex(par_2,"[0-9]+")) %>%
  select(-c("par_1","par_2")) %>%
  pivot_wider(names_from = dim, values_from = med) %>%
  mutate(job1 = job[,1],
         job2 = job[,2],
         job3 = job[,3]) %>%
  pivot_longer(cols = c(job1,job2,job3),names_to = "time",names_prefix = "job") %>%
  mutate(class = as.character(value)) %>%
  select(-value)
med_w <- data.frame(par = rownames(fit_summ), med = fit_summ[,6]) %>%
  filter(grepl("w\\[", par)) %>%
  cSplit("par",direction = "wide") %>%
  mutate(class = stri_extract_first_regex(par_1,"[0-9]+"),
         time = stri_extract_first_regex(par_2,"[0-9]+"),
         dim = stri_extract_first_regex(par_3,"[0-9]+")) %>%
  select(-c("par_1","par_2","par_3"))%>%
  pivot_wider(names_from = dim, values_from = med)
ggplot() +  geom_point(data = med_z, aes(x = `1`, y = `2`, color = class, alpha = 0.5)) +
  geom_point(data = med_w, aes(x = `1`, y = `2`, fill = class),
             shape = 25, size = 4, color = "black")  +
```

```
  facet_wrap(~time)
```



Latent Space Positions with MDS

```
wcols_t1_d1 <-grep("w\\[.,1,1", names(post))
wcols_t1_d2 <-grep("w\\[.,1,2", names(post))
wcols_t2_d1 <-grep("w\\[.,2,1", names(post))
wcols_t2_d2 <-grep("w\\[.,2,2", names(post))
wcols_t3_d1 <-grep("w\\[.,3,1", names(post))
wcols_t3_d2 <-grep("w\\[.,3,2", names(post))
zcols_d1 <- grep("z\\[.*,1\\]", names(post))
zcols_d2 <- grep("z\\[.*,2\\]", names(post))
n_post <- dim(post)[1]
post_mat <- matrix(unlist(post),nrow=n_post,byrow=F)
dist_mat1 <- parDist(x = matrix(c(post_mat[1,wcols_t1_d1],post_mat[1,zcols_d1],
                                  post_mat[1,wcols_t1_d2],post_mat[1,zcols_d2]),
                                ncol = 2), method = "euclidean")
dist_mat2 <- parDist(x = matrix(c(post_mat[1,wcols_t2_d1],post_mat[1,zcols_d1],
                                  post_mat[1,wcols_t2_d2],post_mat[1,zcols_d2]),
                                ncol = 2), method = "euclidean")
dist_mat3 <- parDist(x = matrix(c(post_mat[1,wcols_t3_d1],post_mat[1,zcols_d1],
                                  post_mat[1,wcols_t3_d2],post_mat[1,zcols_d2]),
                                ncol = 2), method = "euclidean")
for (i in 2:n_post) {
  dist1 <- parDist(x = matrix(c(post_mat[i,wcols_t1_d1],post_mat[i,zcols_d1],
```

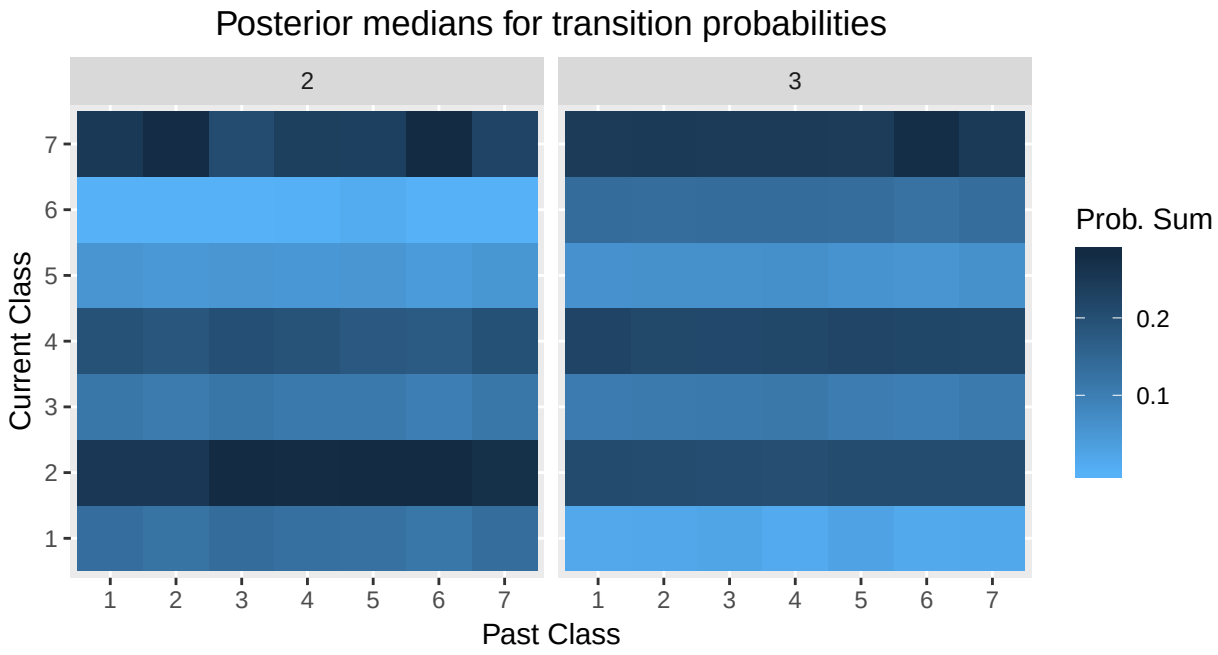
```
                                post_mat[i,wcols_t1_d2],post_mat[i,zcols_d2]),
                              ncol = 2), method = "euclidean")
  dist_mat1 <- dist_mat1 + dist1
  dist2 <- parDist(x = matrix(c(post_mat[i,wcols_t2_d1],post_mat[i,zcols_d1],
                                post_mat[i,wcols_t2_d2],post_mat[i,zcols_d2]),
                              ncol = 2), method = "euclidean")
  dist_mat2 <- dist_mat2 + dist2
  dist3 <- parDist(x = matrix(c(post_mat[i,wcols_t3_d1],post_mat[i,zcols_d1],
                                post_mat[i,wcols_t3_d2],post_mat[i,zcols_d2]),
                              ncol = 2), method = "euclidean")
  dist_mat3 <- dist_mat3 + dist3
}
dist_mat1 <- dist_mat1/n_post
dist_mat2 <- dist_mat2/n_post
dist_mat3 <- dist_mat3/n_post
```

```
mds1 <- cmdscale(dist_mat1, k=2)
mds2 <- cmdscale(dist_mat2, k=2)
mds3 <- cmdscale(dist_mat3, k=2)
#make data frame to plot
plot_mds_z <- data.frame(mds_dim1 = c(mds1[,1],mds2[,1],mds3[,1]),
                         mds_dim2 = c(mds1[,2],mds2[,2],mds3[,2]),
                         time = sort(rep(1:3,N+7)),
                         type = rep(c(rep("w",7),rep("z",N)),3)) %>%
  filter(type == "z")
plot_mds_z$class <- as.factor(c(job[,1],job[,2],job[,3]))
plot_mds_w <- data.frame(mds_dim1 = c(mds1[,1],mds2[,1],mds3[,1]),
                         mds_dim2 = c(mds1[,2],mds2[,2],mds3[,2]),
                         time = sort(rep(1:3,N+7)),
                         type = rep(c(rep("w",7),rep("z",N)),3)) %>%
  filter(type == "w")
plot_mds_w$class <- as.factor(rep(1:7,3))
ggplot() +
  geom_point(data = plot_mds_z, aes(x = mds_dim1, y = mds_dim2, color = class, alpha = 0.5)) +
  geom_point(data = plot_mds_w, aes(x = mds_dim1, y = mds_dim2, fill = class),
             shape = 25, size = 4, color = "black")  +
  facet_wrap(~time)
```

Probability transition plots - Posterior medians

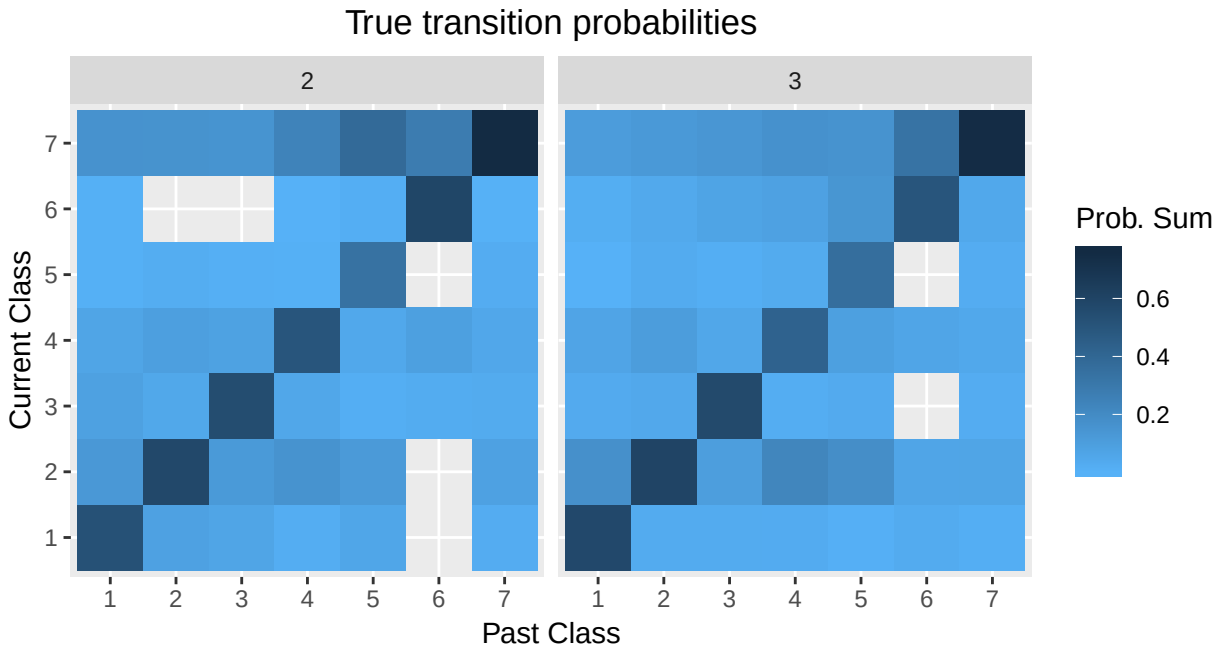
```
med_p <- data.frame(par = rownames(fit_summ), med = fit_summ[,6]) %>%
  filter(grepl("p\\[", par)) %>%
  cSplit("par",direction = "wide") %>%
  mutate(worker = stri_extract_first_regex(par_1,"[0-9]+"),
         time = stri_extract_first_regex(par_2,"[0-9]+"),
         class = stri_extract_first_regex(par_3,"[0-9]+")) %>%
  select(-c("par_1","par_2","par_3")) %>%
  filter(time %in% c(2,3)) %>%
  mutate(past_class = c(rep(job[,2],each=7),rep(job[,3],each=7)))
trans_probs <- med_p %>% group_by(time,past_class) %>%
  mutate(tot_sum = sum(med),
         prop = med/tot_sum) %>%
```

```
  group_by(time,class,past_class) %>%
  summarise(prob_sum = sum(prop)) %>%
  arrange(past_class)
ggplot(trans_probs, aes(x = as.factor(past_class), y = as.factor(class), fill = prob_sum)) +
  geom_tile() + facet_grid(cols = vars(time)) +
  xlab("Past Class") + ylab("Current Class") + labs(fill='Prob. Sum') +
  ggtitle("Posterior medians for transition probabilities") +
  theme(aspect.ratio = 1) + theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_continuous(high = "#132B43", low = "#56B1F7")
```



Probability transition plots - Data

```
real_trans <- data.frame(past = c(job[,1],job[,2]), current = c(job[,2],job[,3]),
                         time = c(rep(2,N),rep(3,N))) %>%
  group_by(past,time) %>%
  mutate(tot_sum = n(),
         prop = 1/tot_sum) %>%
  group_by(past,current,time) %>%
  summarise(prob_sum = sum(prop)) %>%
  arrange(past,time)
ggplot(real_trans, aes(x = as.factor(past), y = as.factor(current), fill = prob_sum)) +
  geom_tile() + facet_grid(cols = vars(time)) +
  xlab("Past Class") + ylab("Current Class") + labs(fill='Prob. Sum') +
  ggtitle("True transition probabilities") +
  theme(aspect.ratio = 1) + theme(plot.title = element_text(hjust = 0.5)) +
  scale_fill_continuous(high = "#132B43", low = "#56B1F7")
```


True transition probabilities
2
3
Current Class
Past Class
1
2
3
4
5
6
7
Prob. Sum
0.6
0.4
0.2