NFL Attendance in 2020

Cason Wight

2023-03-17

Prep Work for Analysis

Load all tools and functions

```r
source("utils/eda.R")
```

```
## -- Attaching packages ---------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.1.0
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
source("utils/get_data.R")
```

```
##
## Attaching package: 'rvest'
##
## The following object is masked from 'package:readr':
##
##     guess_encoding
##
## Using libcurl 7.84.0 with Schannel
##
## Attaching package: 'curl'
##
## The following object is masked from 'package:readr':
##
##     parse_date
```

```r
source("utils/mcmc_diagnostics.R")
source("utils/stan.R")
source("models/bayesian.R")
```

```
## Loading required package: StanHeaders
##
## rstan version 2.26.13 (Stan version 2.26.1)
##
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
## For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
## change `threads_per_chain` option:
## rstan_options(threads_per_chain = 1)
##
## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file
##
## Attaching package: 'rstan'
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
source("results/calculations.R")
source("results/plots.R")
```

Load data

```r
attendance <- get_attendance_data() %>%
  mutate(home_team = str_replace(home_team, "Redskins", "Commanders"))
print(attendance %>% head())
```

```
##   X year home_team total_attendance num_games num_games_won prev_attendance
## 1 1 2001     49ers         67469.50        16            12        67745.50
## 2 2 2001     Bears         66944.00        16            13        66944.00
## 3 3 2001   Bengals         56867.25        16             6        58749.00
## 4 4 2001     Bills         63092.00        16             3        70086.88
## 5 5 2001   Broncos         75035.38        16             8        75505.25
## 6 6 2001    Browns         72886.75        16             7        72693.00
##   prev_games_won
## 1              6
## 2              5
## 3              4
## 4              8
## 5             11
## 6              3
```

```r
attendance_2020 <- get_2020_attendance_data() %>%
  mutate(team_name = str_replace(team_name, "Redskins", "Commanders"))
print(attendance_2020 %>% head())
```

```
##   X          city team_name      Home
## 1 1 San Francisco     49ers     0.000
## 2 2       Chicago     Bears     0.000
## 3 3    Cincinnati   Bengals  4185.312
## 4 4       Buffalo     Bills     0.000
## 5 5        Denver   Broncos  1320.125
## 6 6     Cleveland    Browns  5027.500
```

```r
prices_2019 <- get_average_ticket_price_data()
print(prices_2019 %>% head())
```

```
##   X team_name avg_cost_2019
## 1 1     49ers           178
## 2 2     Bears           376
## 3 3   Bengals           114
## 4 4     Bills           105
```

```
## 5 5    Broncos          278
## 6 6     Browns          190
```

```
team_names <- attendance_2020 %>%
  pull(team_name) %>%
  unique() %>%
  as.character()
print(team_names %>% head())
```

```
## [1] "49ers"   "Bears"   "Bengals" "Bills"   "Broncos" "Browns"
```

```
team_cities <- sapply(team_names, function(name) {
  attendance_2020 %>%
    filter(team_name == name) %>%
    pull(city) %>%
    .[1]
})
print(team_cities %>% head())
```

```
##             49ers            Bears          Bengals            Bills          Broncos
## "San Francisco"        "Chicago"     "Cincinnati"        "Buffalo"         "Denver"
##            Browns
##      "Cleveland"
```

```
rev_data <- get_revenue_data()
print(rev_data %>% head())
```

```
##    Year Total_Revenue Label
## 1 2010          8.35
## 2 2011          8.82
## 3 2012          9.17
## 4 2013          9.58
## 5 2014         11.09
## 6 2015         12.16
```

```
rev_team_data <- get_team_revenue_data(team_names, team_cities)
print(rev_team_data %>% head())
```

```
##   X year rev team_name  label
## 1 1 2019 492      49ers $492 M
## 2 2 2018 470      49ers
## 3 3 2017 458      49ers
## 4 4 2016 446      49ers
## 5 5 2015 427      49ers
## 6 6 2014 270      49ers
```

Run models (may take several hours, skip to load models if already done)

```
fit_samps <- fit_bayesian_mods() %>%
  save_fit_tight_priors() %>%
  save_samps_tight_priors() %>%
  downsample_samps()
```

Load models (much faster if models have already been run and samples saved, can be skipped if running model in above chunk)

```r
fit_samps <- load_bayesian_samps() %>% downsample_samps()
```

Load other model metadata

```r
prior <- get_priors()[[1]]
print(prior)
```

```
## $mu_a
## [1] 30000
##
## $sigma_a
## [1] 20000
##
## $mu_b
## [1] 0.7
##
## $sigma_b
## [1] 0.5
##
## $mu_theta
## [1] 1000
##
## $sigma_theta
## [1] 1000
##
## $a_sigma
## [1] 0.4444444
##
## $b_sigma
## [1] 4.444444e-05
##
## $lambda
## [1] 8000
##
## $eta
## [1] 0.1
```

```r
var_names <- fit_samps %>% names() %>% .[1:6]
print(var_names)
```

```
## [1] "overall_alpha" "overall_beta"  "theta"         "sigma"
## [5] "alphas"        "betas"
```

Model:

$$\text{Est 2020 Attendance}_{\text{team}} \sim \mathcal{N}\left(\mu_{\text{team}}, \sigma^2\right)$$

$$\mu_{\text{team}} = \alpha_{\text{team}}$$
$$+ \beta_{\text{team}} \times 2019 \text{ Year Attendance}$$
$$+ \theta \times 2019 \text{ Num Games Won}$$

$$\sigma \sim \text{Gamma}(\alpha_\sigma, \beta_\sigma)$$
$$\alpha_{\text{team}} \sim \mathcal{N}(\alpha, \lambda^2)$$
$$\alpha \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^2)$$
$$\beta_{\text{team}} \sim \mathcal{N}(\beta, \eta^2)$$
$$\beta \sim \mathcal{N}(\mu_\beta, \sigma_\beta^2)$$
$$\theta \sim \mathcal{N}(\mu_\theta, \sigma_\theta^2)$$

Exploratory Data Analysis

Who has the most attendees at their home games?

```
plot_avg_attendance_all_years(attendance, "All teams")
```


Average Home–Game Attendance from 2000 to 2019
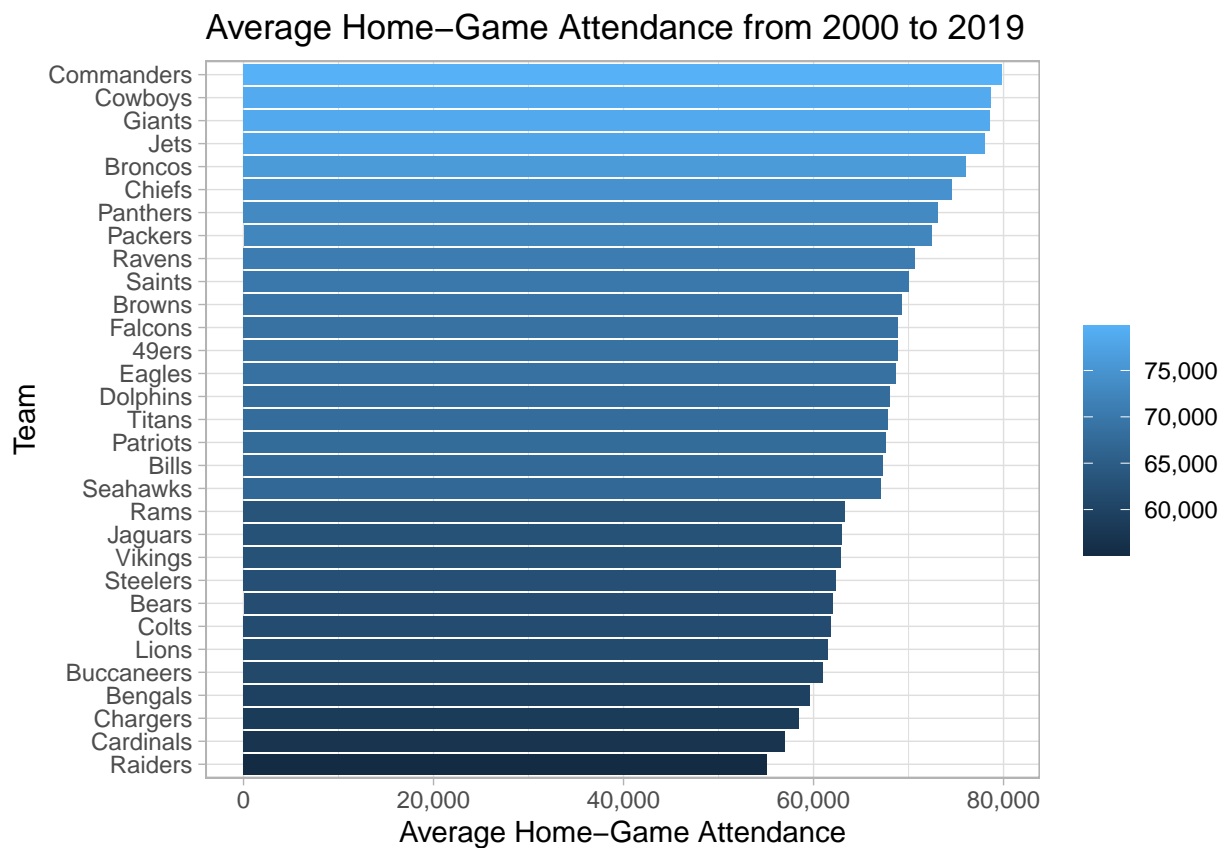
Commanders, Cowboys, and Giants have the highest attendance. Raiders, Cardinals, and Chargers have the lowest.

Who wins the most?

```
plot_avg_win_rate_all_years(attendance, "All teams")
```

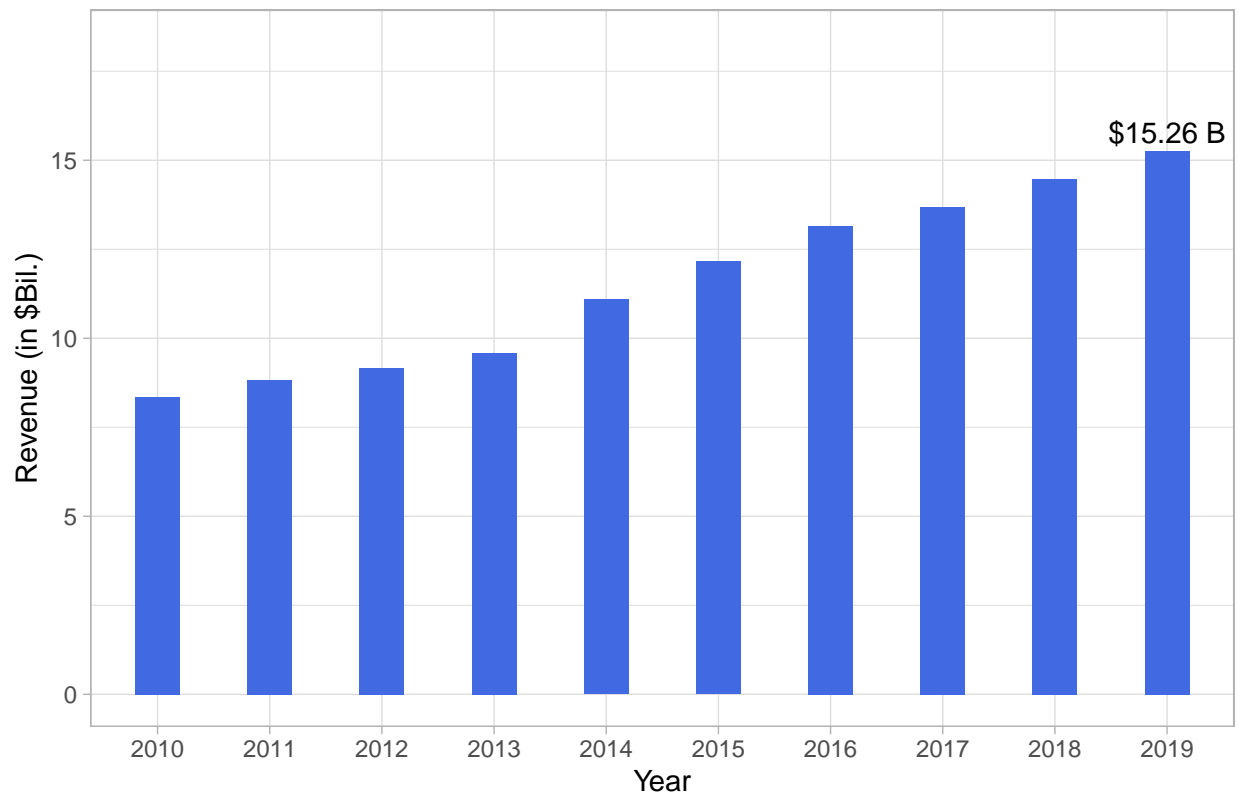## Regular Game Wins (2000 to 2019)
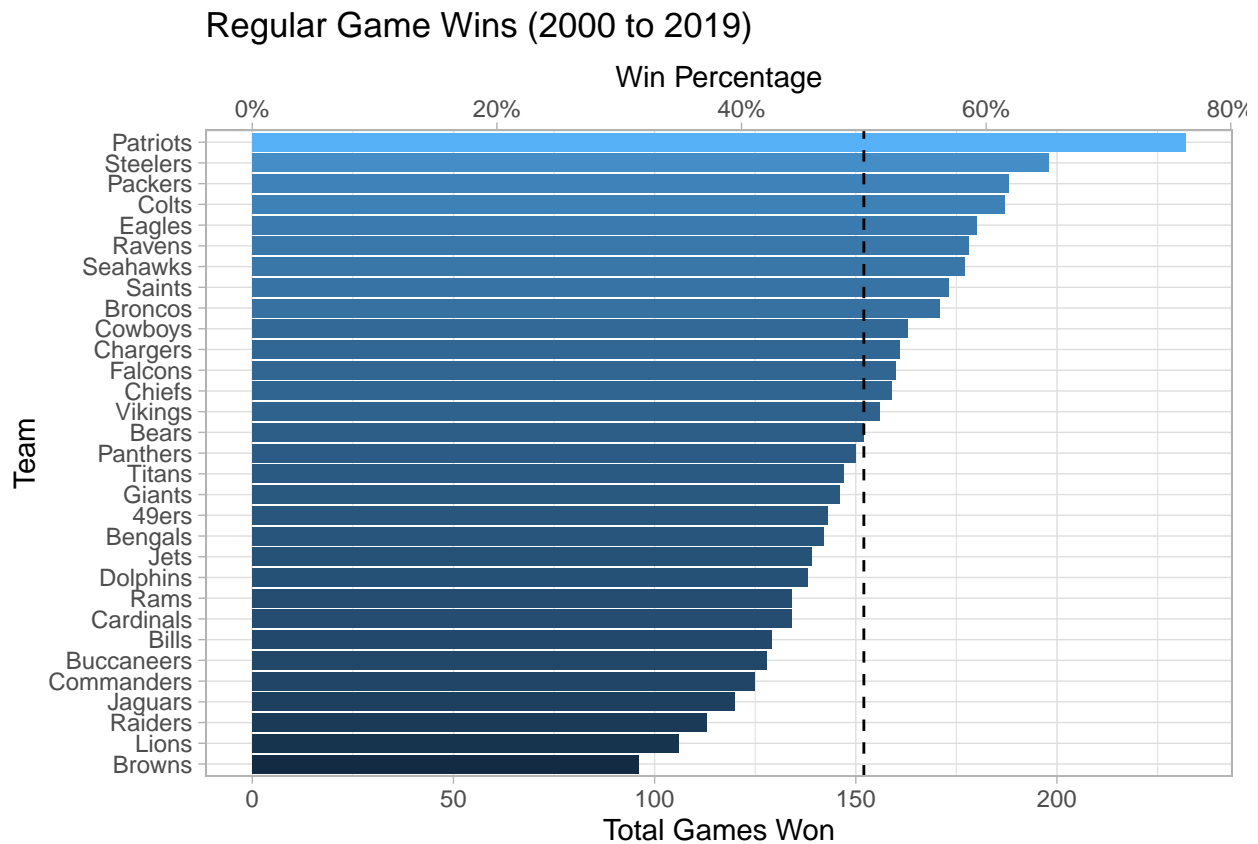
Patriots, Steelers, and Packers win the most. Browns, Lions, and Raiders lose the most.

How much money does the NFL make?

```
plot_yearly_revenue(rev_data, rev_team_data, "All teams")
```

Total NFL Revenue

There is a roughly linear trend, increasing by about $1B every year. In 2019, NFL revenue totaled $15.26B.
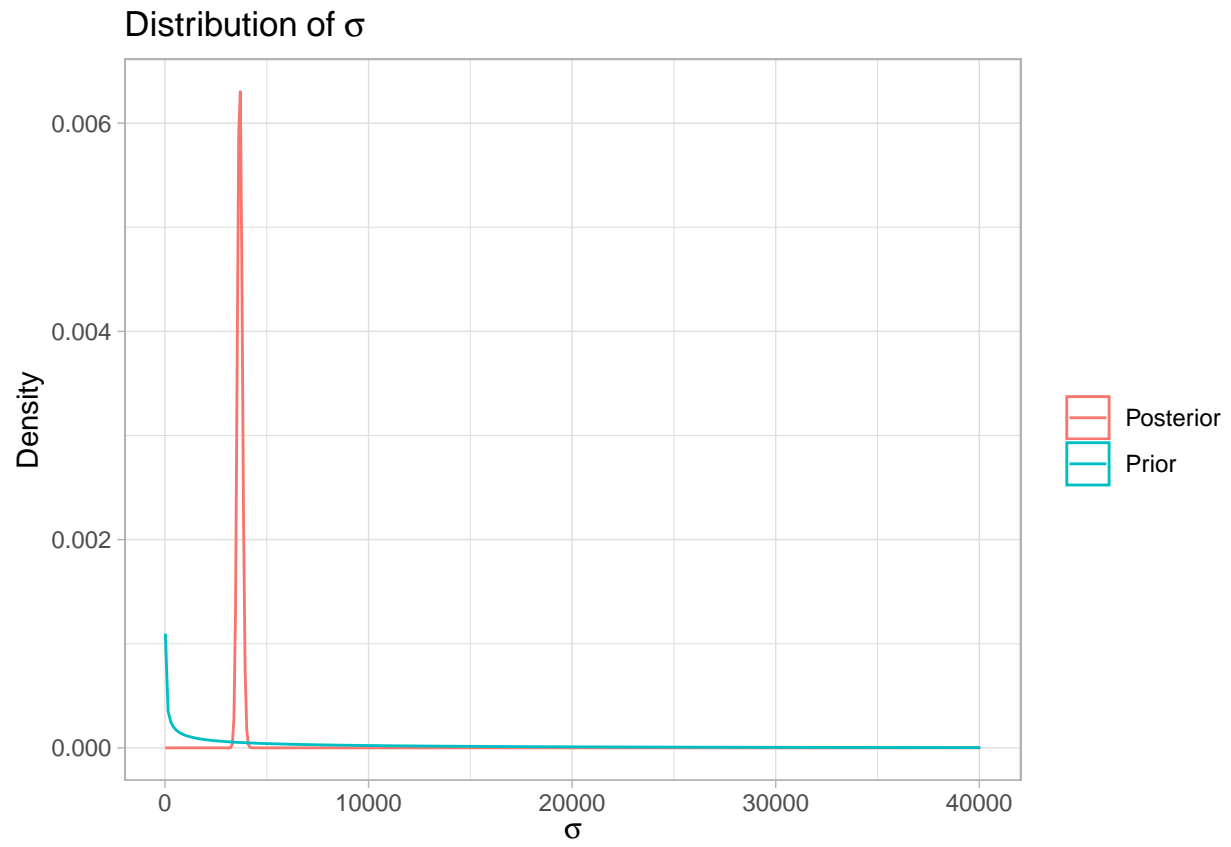
Parameter modeling

What is the prior and posterior predicted distribution of the parameters?

```r
for (var in c("overall_alpha", "overall_beta", "theta", "sigma", "alphas", "betas")){
  plt <- plot_distribution(fit_samps, prior, var, team_names, "All teams")
  print(plt)
}
```
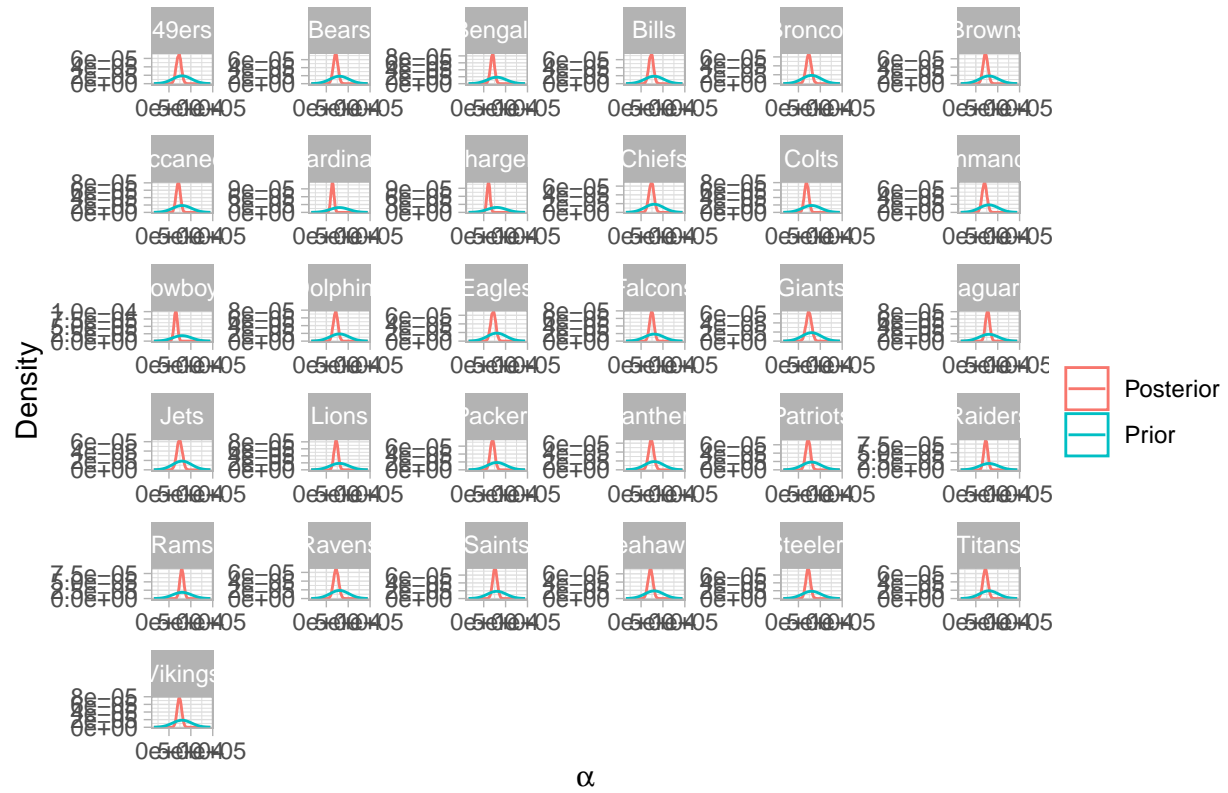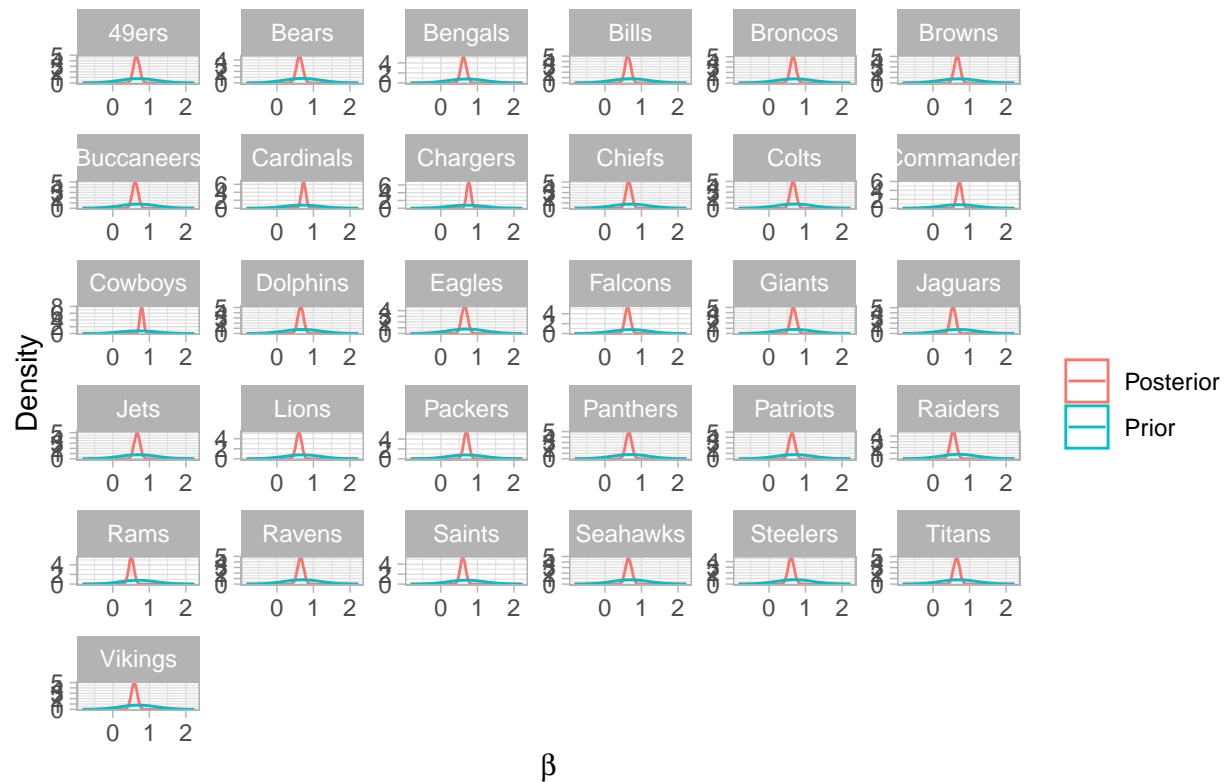
Distribution of α

Distribution of β

Distribution of θ

Distribution of σ

Legend:
- Posterior
- Prior
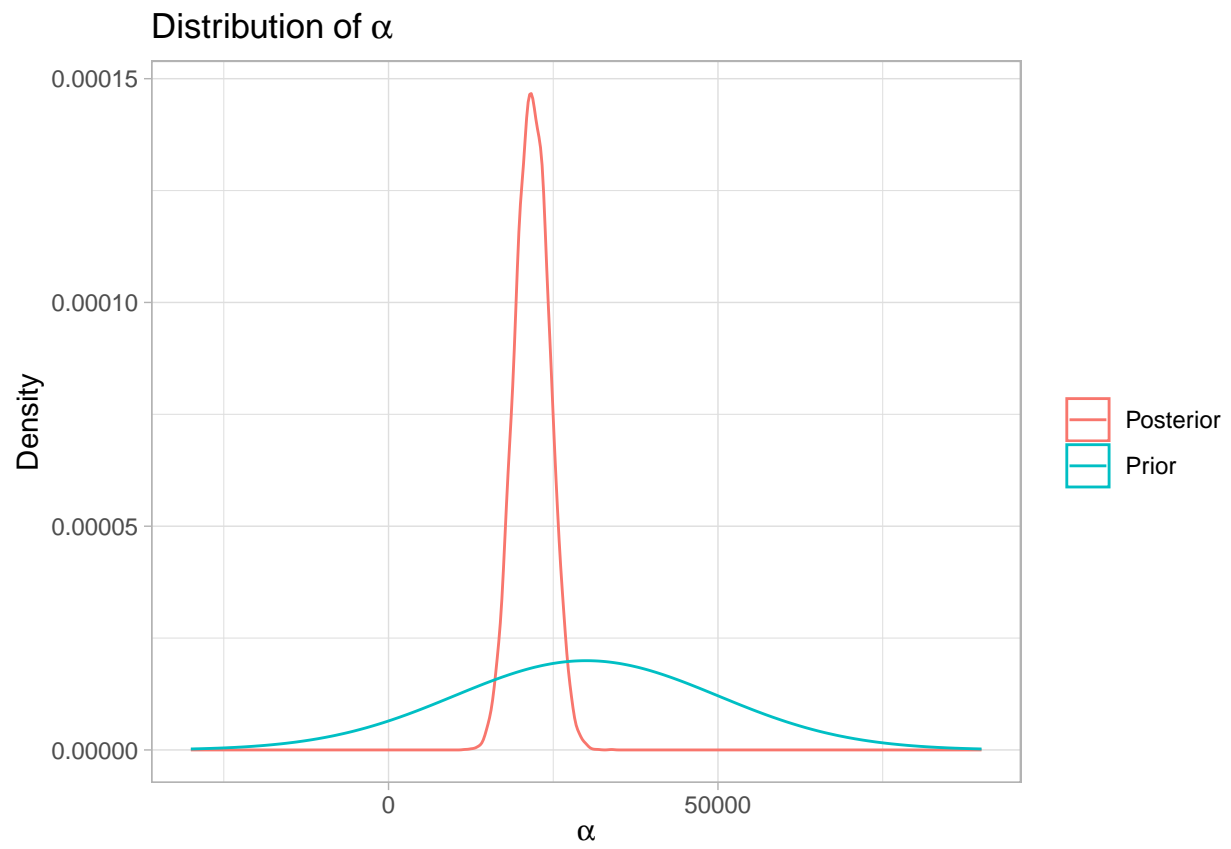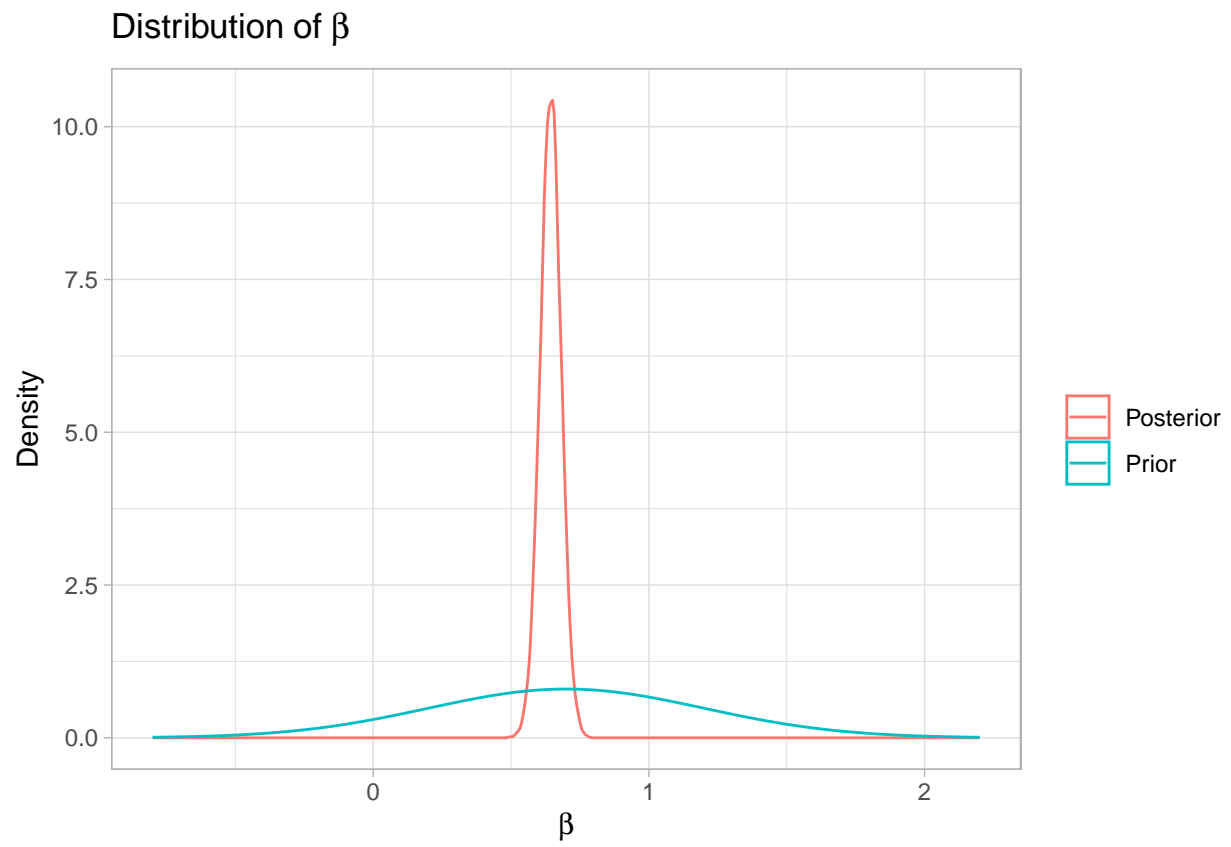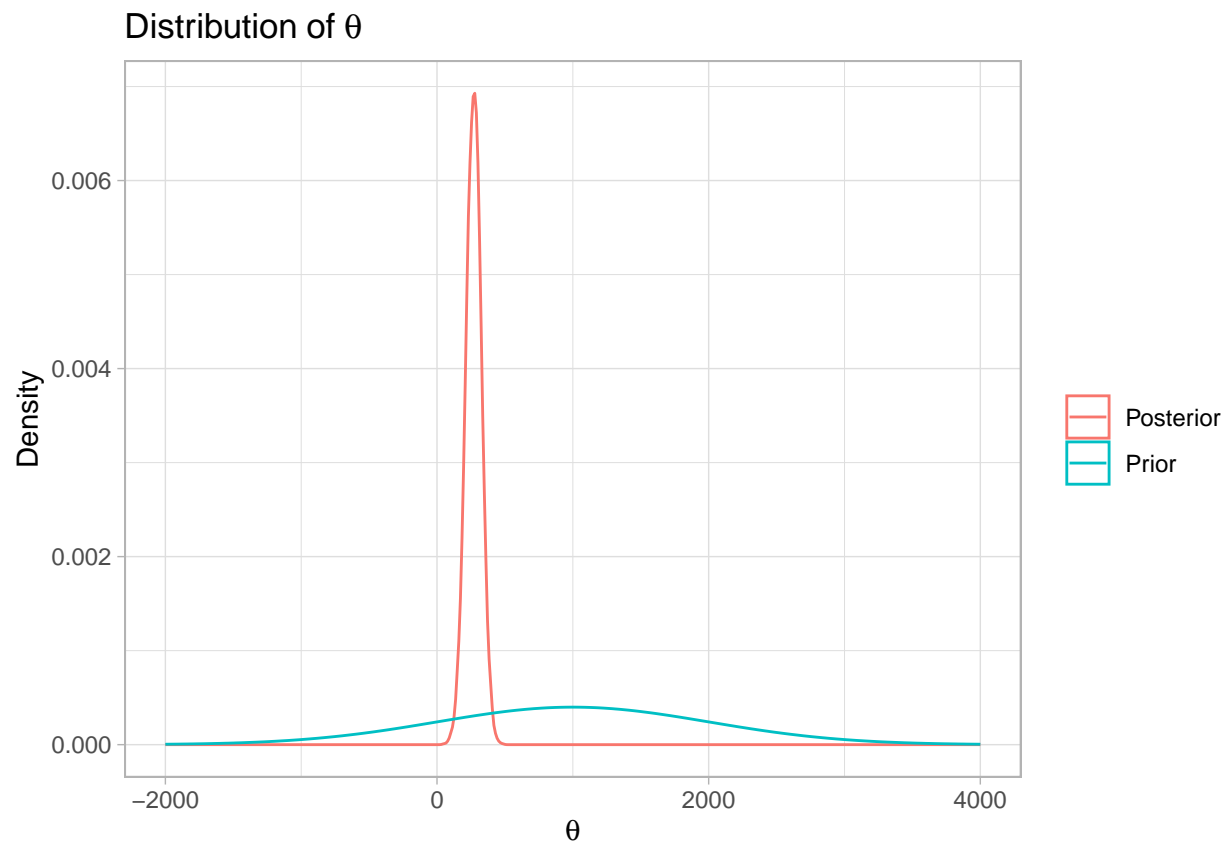
Distributions of α

Distributions of β

**Do the MCMC diagnoistics look healthy (was this a good fit)?**

```r
for (var in c("overall_alpha", "overall_beta", "theta", "sigma", "alphas", "betas")) {
  neff <- get_effective_sample_size(fit_samps, var, team_names, team_name="All teams")
  rhat <- get_rhat(fit_samps, var, team_names, team_name="All teams")
  trace_plt <- get_trace_plot(fit_samps, var, team_names, team_name="All teams")
  acf_plt <- get_acf_plot(fit_samps, var, team_names, team_name="All teams")

  print(var)
  print("Effective sample size:")
  print(neff)
  print("Gelman statistic:")
  print(rhat)
  print(trace_plt)
  print(acf_plt)
  print("")
}
```

```
## [1] "overall_alpha"
## [1] "Effective sample size:"
## overall_alpha
##      9361.625
## [1] "Gelman statistic:"
## overall_alpha
##       1.00058
```
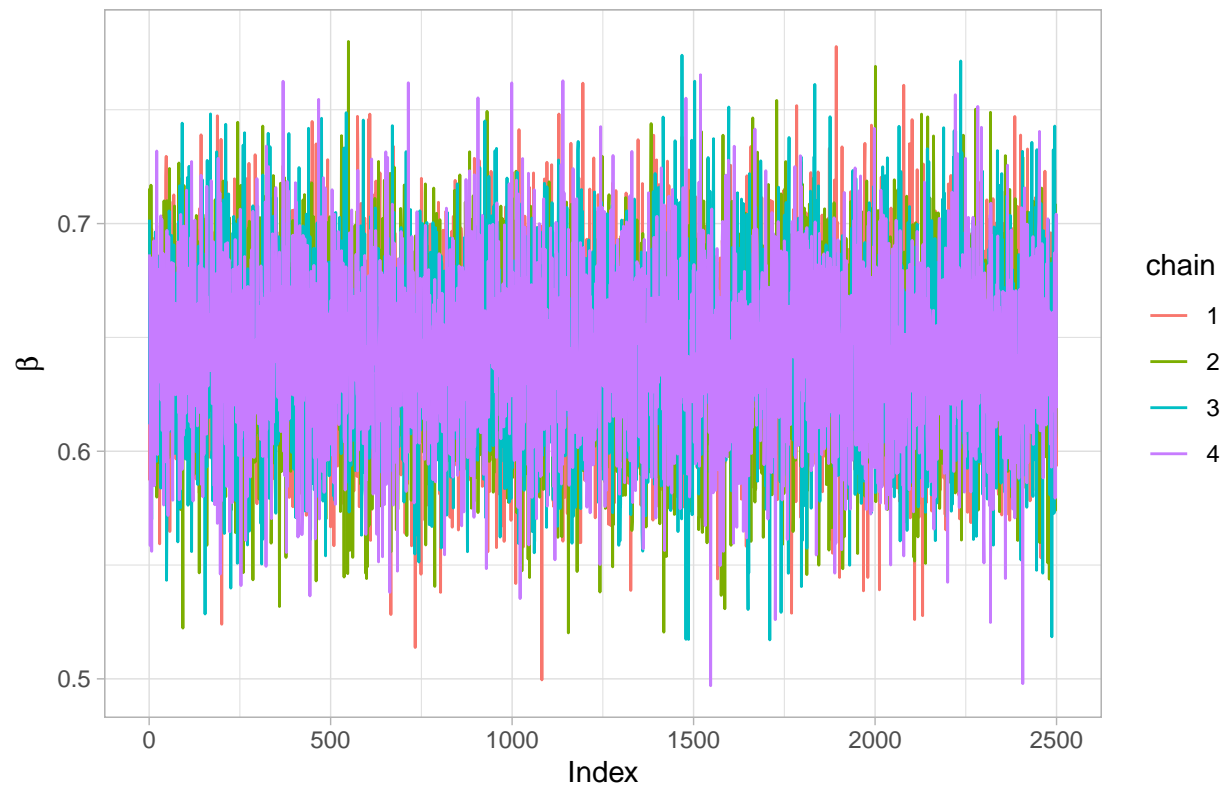
Trace Plot of α
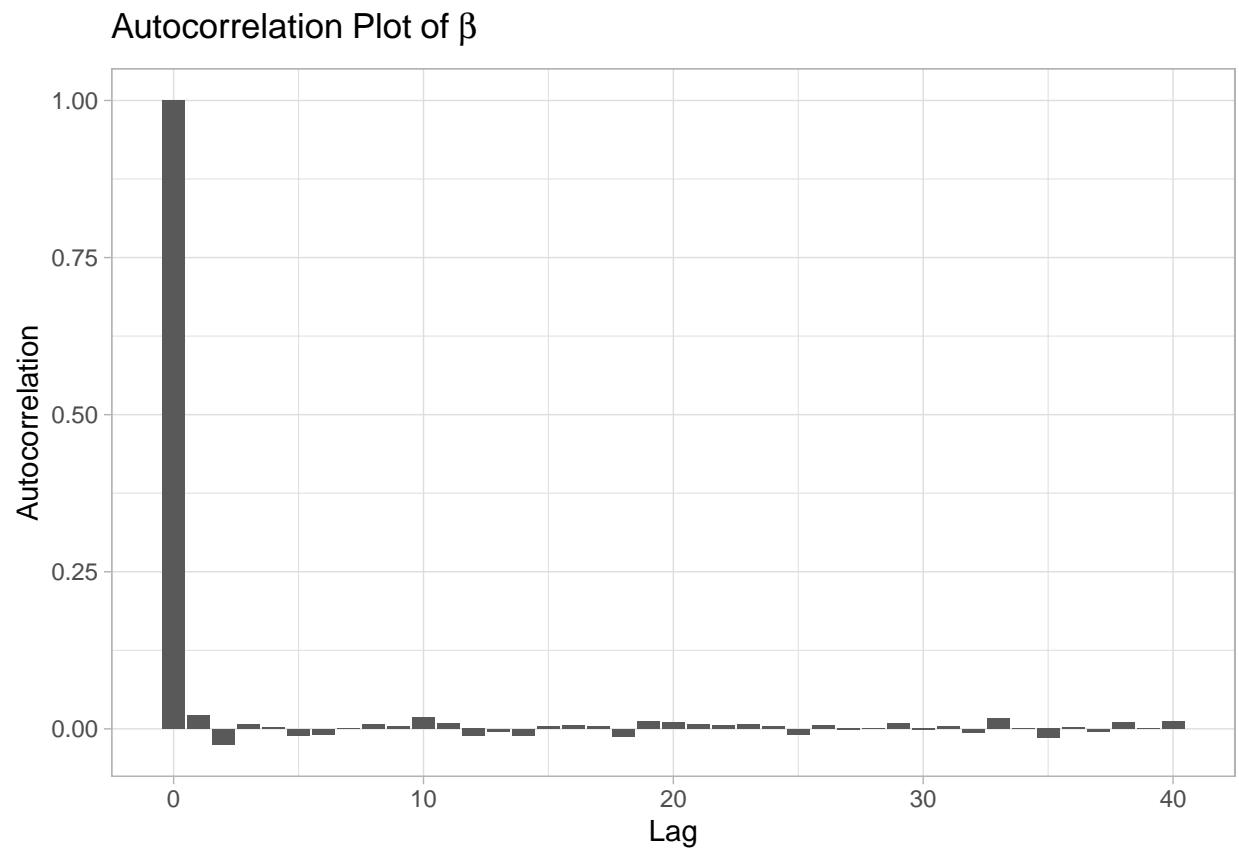
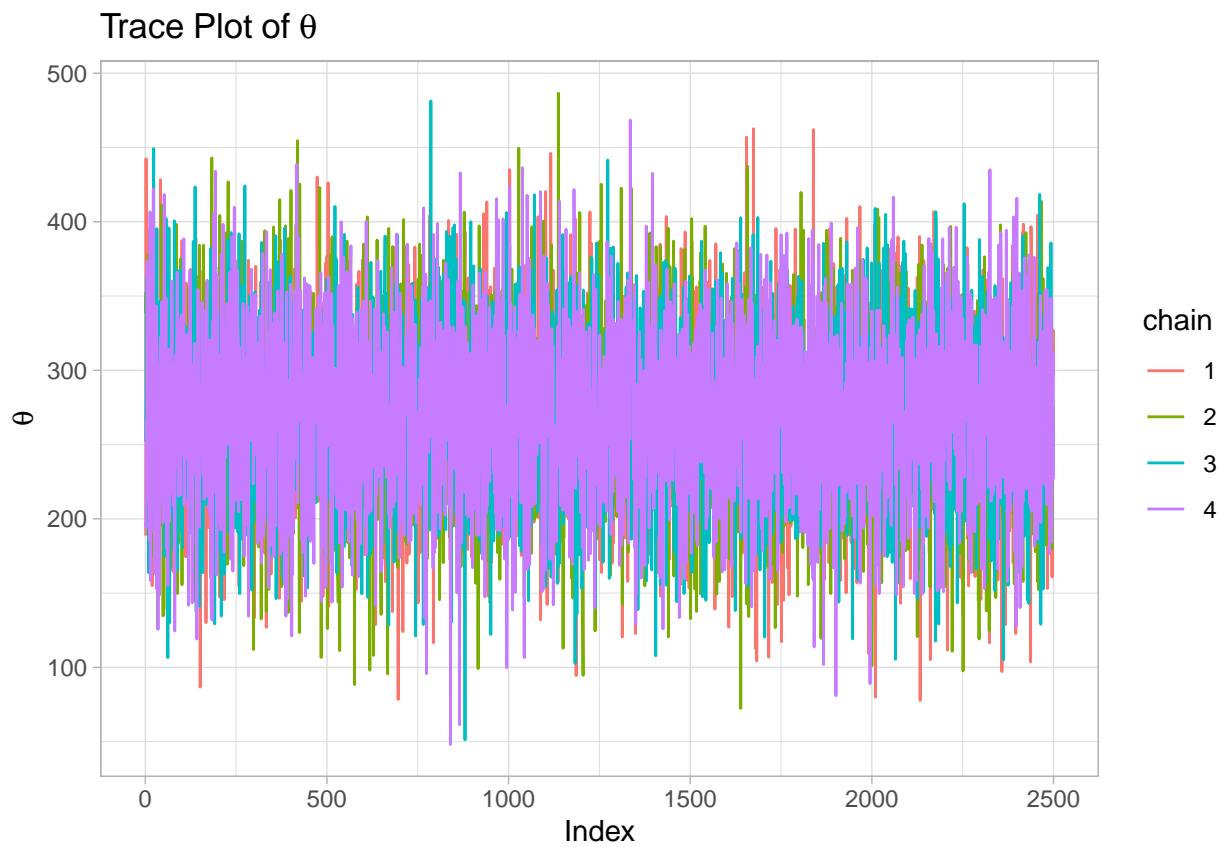## Autocorrelation Plot of α
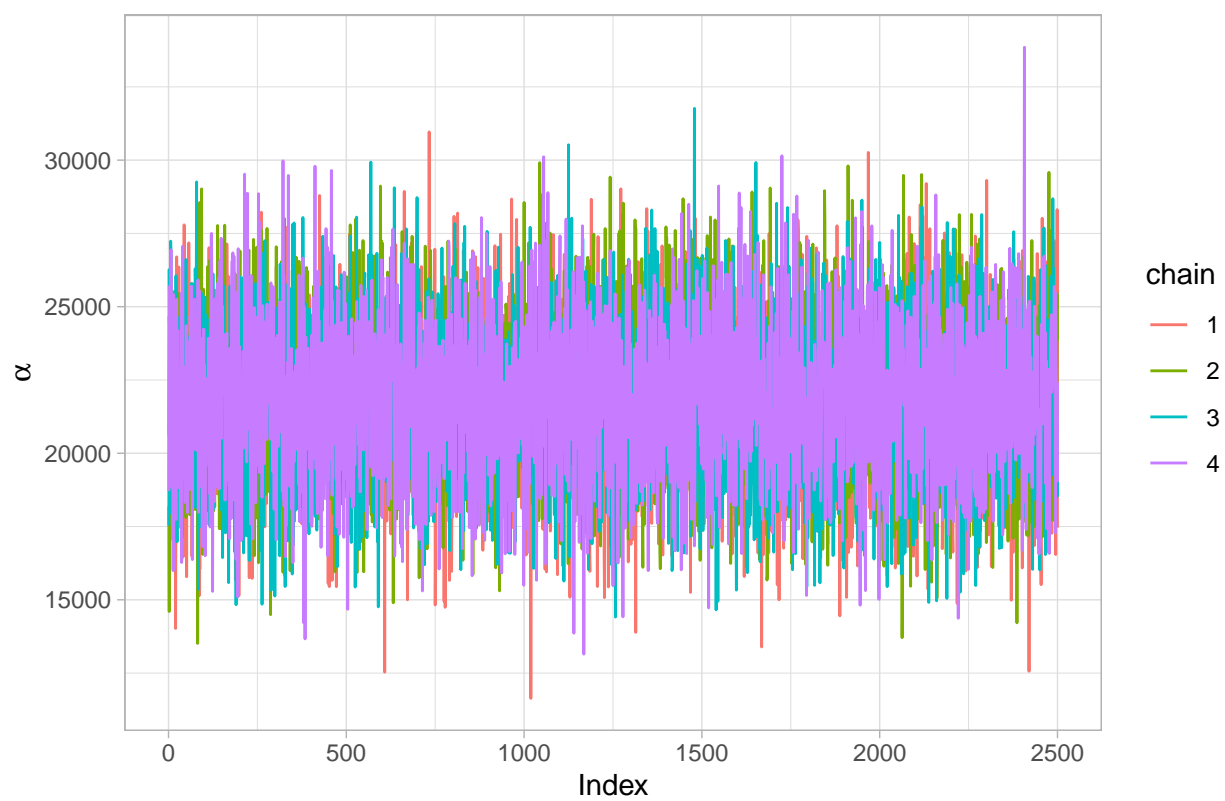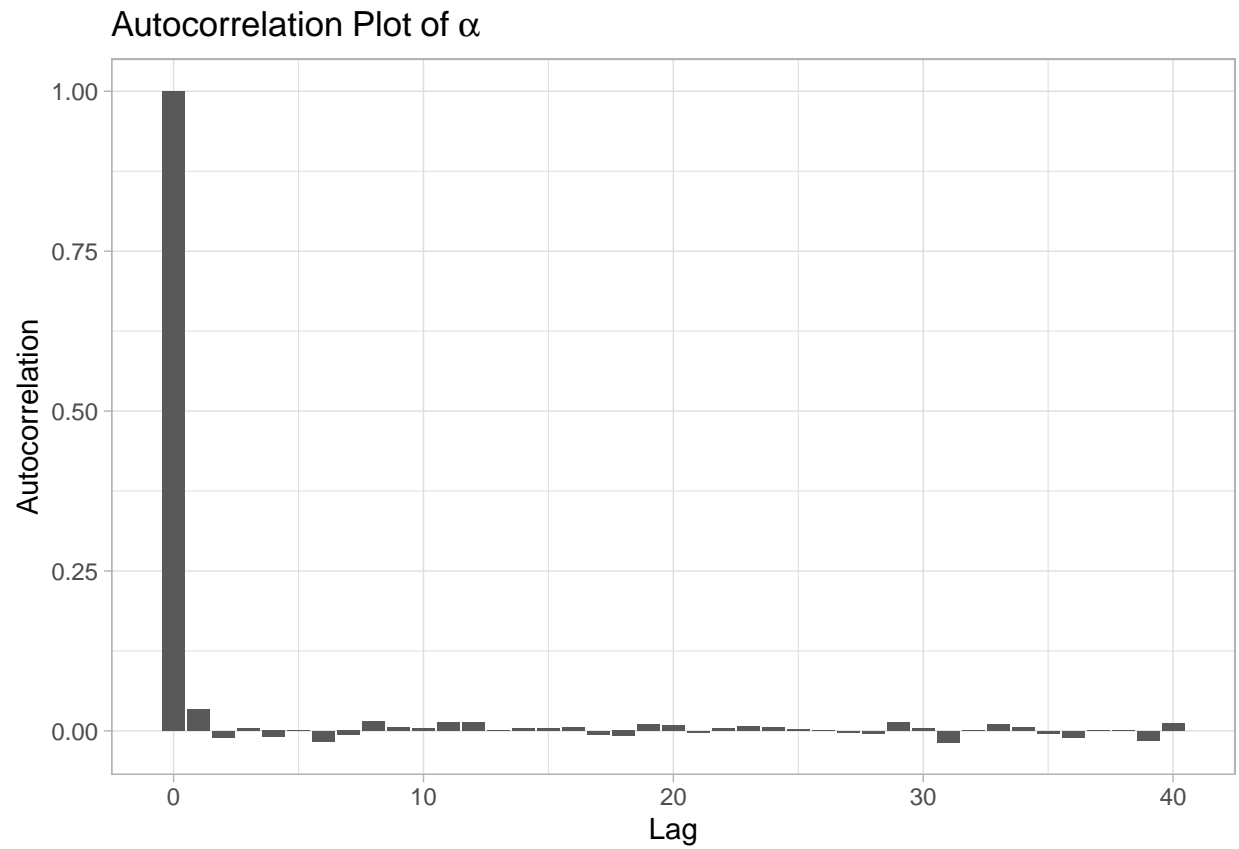

```
## [1] ""
## [1] "overall_beta"
## [1] "Effective sample size:"
## overall_beta
##     10091.17
## [1] "Gelman statistic:"
## overall_beta
##     1.000223
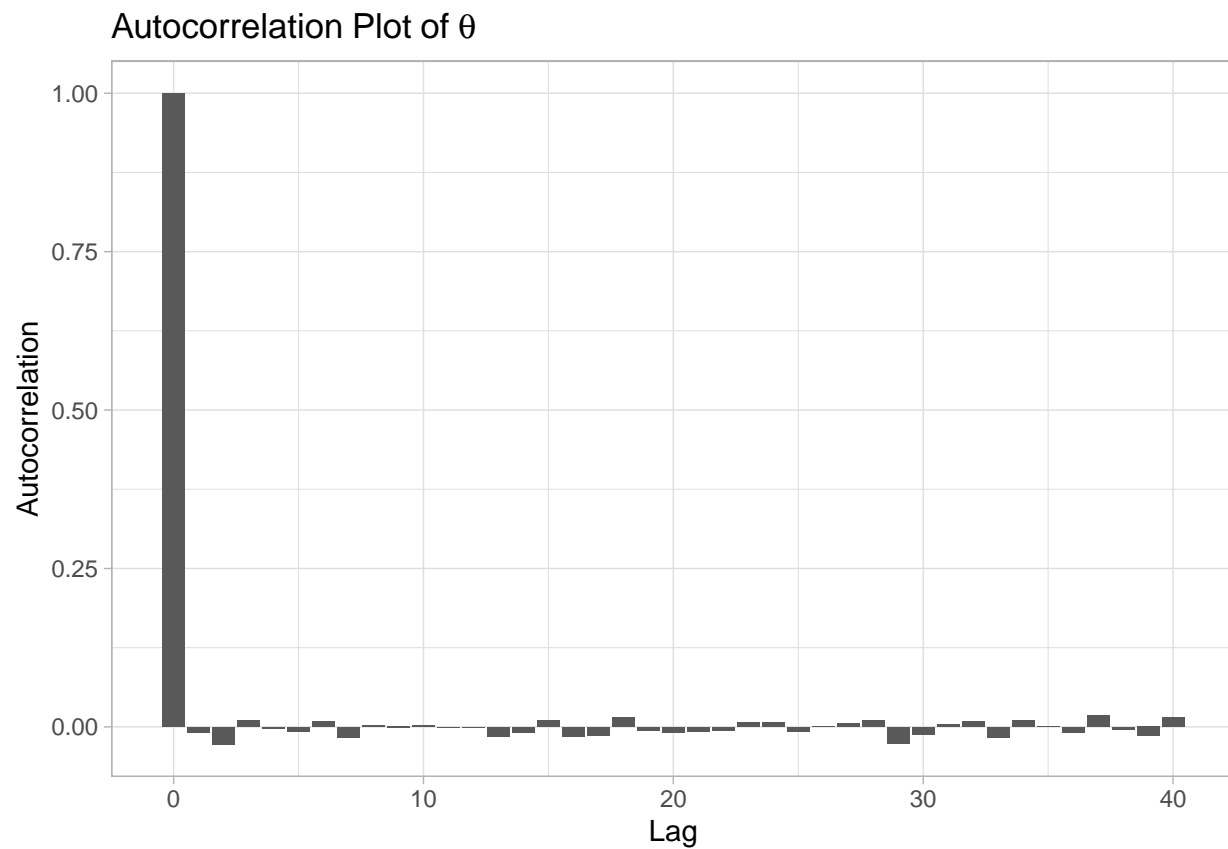```

Trace Plot of β

16

## Autocorrelation Plot of β
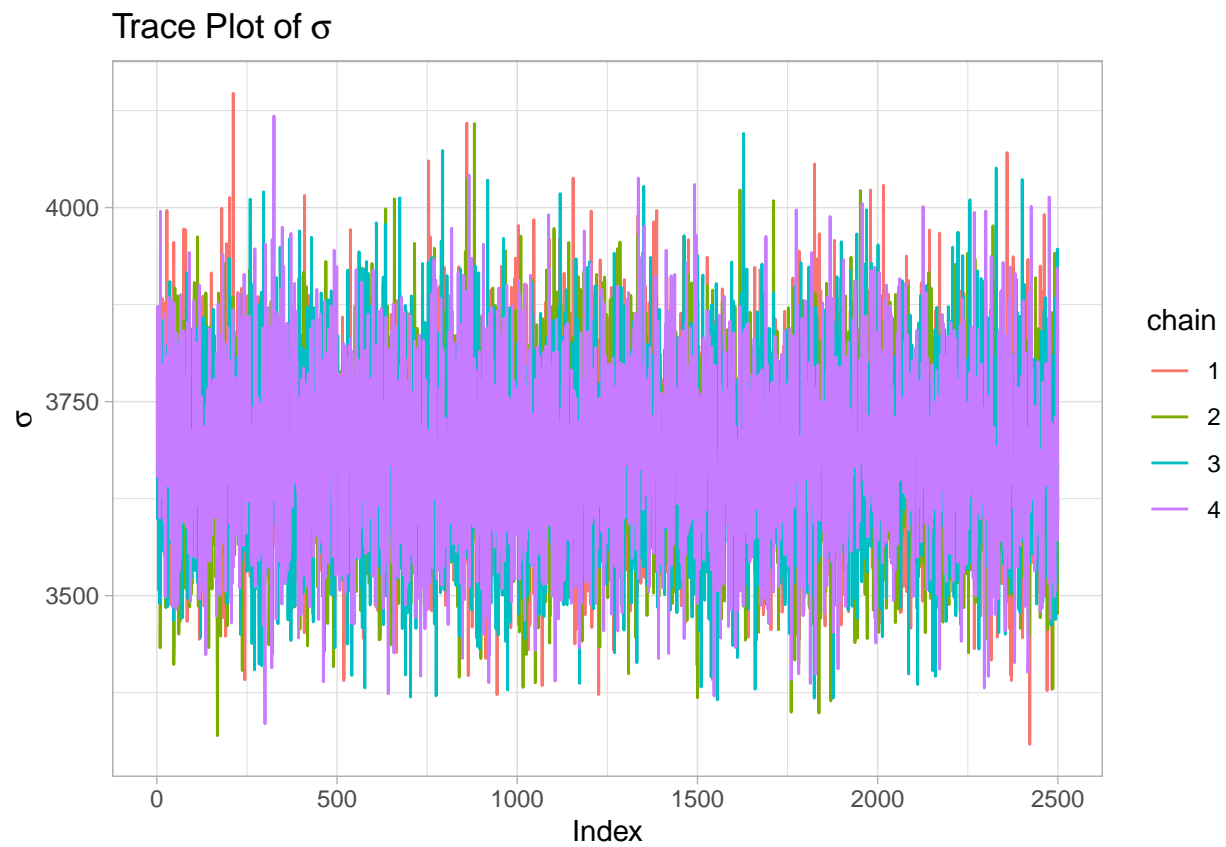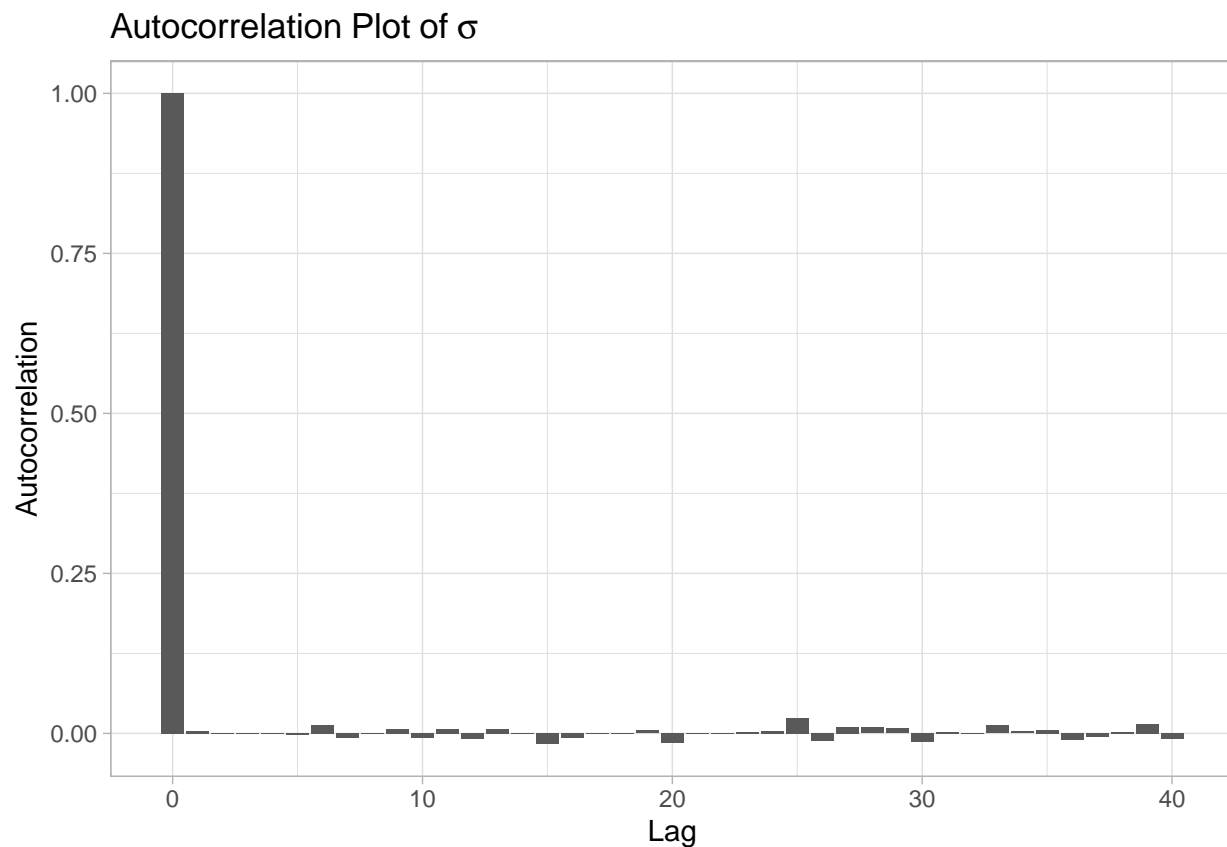


```
## [1] ""
## [1] "theta"
## [1] "Effective sample size:"
##    theta
## 10759.11
## [1] "Gelman statistic:"
##    theta
## 1.000634
```

Trace Plot of θ

Autocorrelation Plot of θ


```
## [1] ""
## [1] "sigma"
## [1] "Effective sample size:"
## sigma
## 10000
## [1] "Gelman statistic:"
##    sigma
## 1.000194
```

Trace Plot of σ

Autocorrelation Plot of σ



```
## [1] ""
## [1] "alphas"
## [1] "Effective sample size:"
##      alpha_49ers       alpha_Bears     alpha_Bengals       alpha_Bills
##        10000.000         10097.454         10000.000         10000.000
##     alpha_Broncos      alpha_Browns  alpha_Buccaneers  alpha_Cardinals
##        10000.000         10000.000         10000.000          9952.882
##    alpha_Chargers       alpha_Chiefs       alpha_Colts      alpha_Cowboys
##        10013.984         10000.000         10424.654         10289.755
##    alpha_Dolphins       alpha_Eagles     alpha_Falcons       alpha_Giants
##        11525.717          9561.790          9828.234         10695.506
##     alpha_Jaguars        alpha_Jets       alpha_Lions      alpha_Packers
##        10000.000         10000.000         10000.000         10638.706
##    alpha_Panthers    alpha_Patriots     alpha_Raiders         alpha_Rams
##        10000.000         10000.000         11226.537         10000.000
##      alpha_Ravens alpha_Commanders       alpha_Saints     alpha_Seahawks
##        10000.000         10129.888         10000.000         10000.000
##    alpha_Steelers      alpha_Titans      alpha_Vikings
##        10000.000         10000.000          9680.110
## [1] "Gelman statistic:"
##      alpha_49ers       alpha_Bears     alpha_Bengals       alpha_Bills
##        1.0001345         1.0003879         1.0003887         1.0000405
##     alpha_Broncos      alpha_Browns  alpha_Buccaneers  alpha_Cardinals
##        1.0002385         1.0001435         1.0003167         0.9999021
##    alpha_Chargers       alpha_Chiefs       alpha_Colts      alpha_Cowboys
##        0.9999854         1.0001012         0.9998713         1.0000578
```

```
##     alpha_Dolphins      alpha_Eagles     alpha_Falcons      alpha_Giants
##          1.0002033         1.0000587         1.0000391         1.0002505
##      alpha_Jaguars        alpha_Jets       alpha_Lions     alpha_Packers
##          0.9999859         1.0002011         1.0003116         1.0000384
##     alpha_Panthers    alpha_Patriots     alpha_Raiders        alpha_Rams
##          1.0000280         1.0006721         0.9999788         1.0000967
##       alpha_Ravens alpha_Commanders      alpha_Saints    alpha_Seahawks
##          1.0013469         0.9999963         1.0000527         1.0003684
##      alpha_Steelers      alpha_Titans     alpha_Vikings
##          0.9998999         1.0001524         1.0003545
```
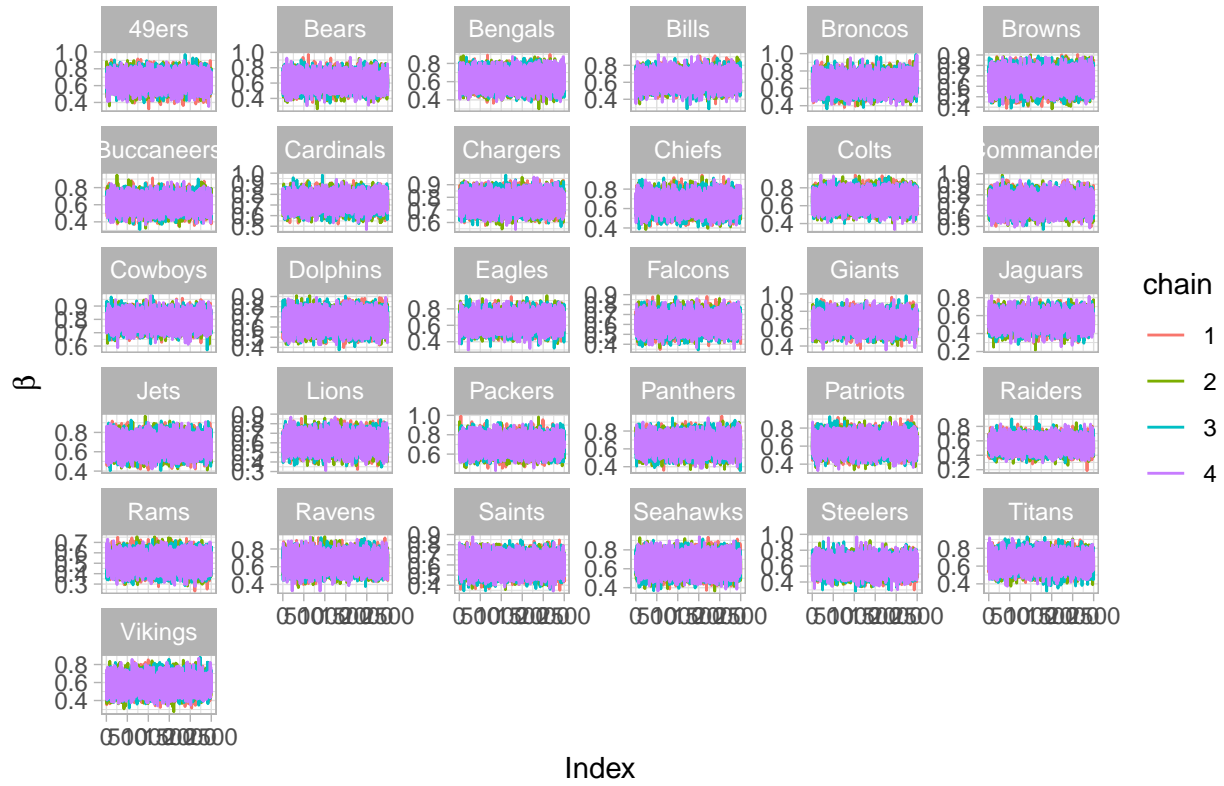
Trace Plots of α

Index
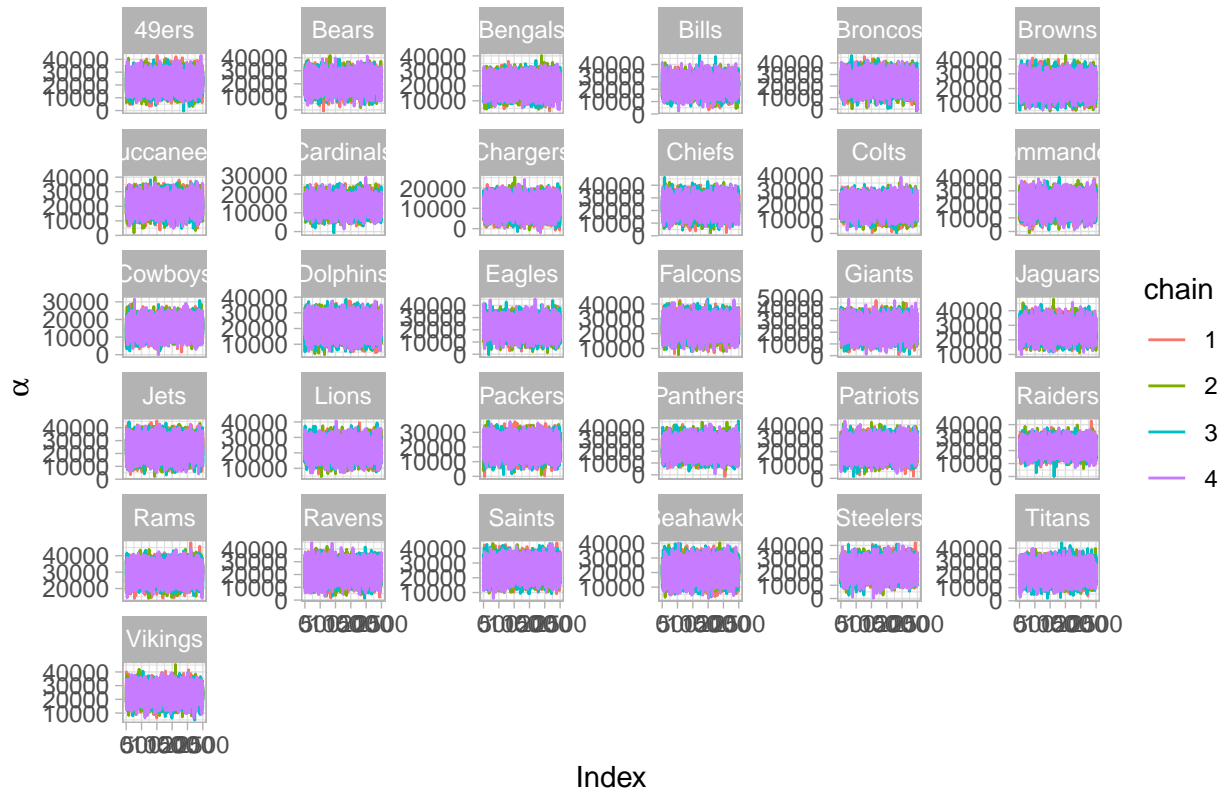
Autocorrelation Plots of α
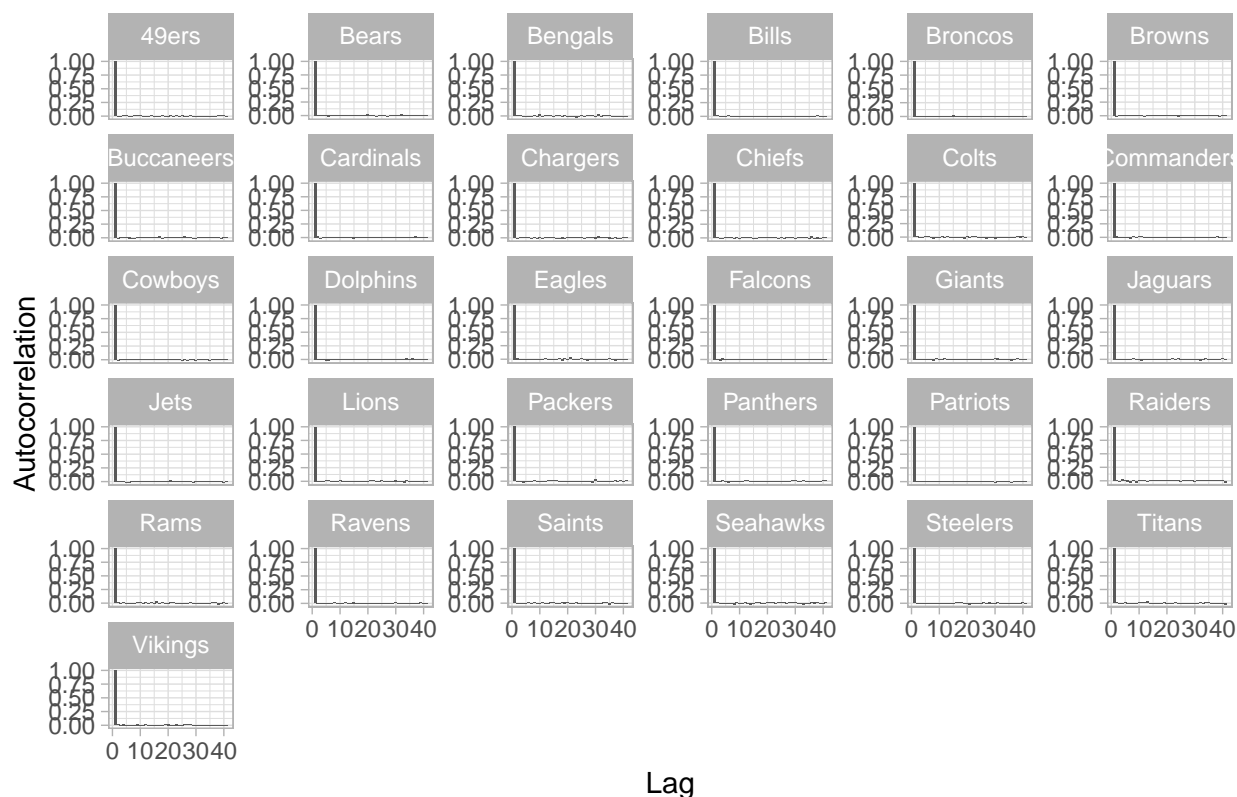


```
## [1] ""
## [1] "betas"
## [1] "Effective sample size:"
##       beta_49ers      beta_Bears    beta_Bengals      beta_Bills     beta_Broncos
##       10000.000       10084.855       10000.000       10000.000       10000.000
##     beta_Browns  beta_Buccaneers   beta_Cardinals    beta_Chargers      beta_Chiefs
##       10000.000       10000.000        9711.032       10313.638       10000.000
##      beta_Colts     beta_Cowboys    beta_Dolphins      beta_Eagles     beta_Falcons
##       10000.000       10391.195       11669.923        9583.362       10160.598
##     beta_Giants     beta_Jaguars        beta_Jets       beta_Lions     beta_Packers
##       10717.587       10000.000       10000.000       10000.000       10674.189
##   beta_Panthers    beta_Patriots     beta_Raiders        beta_Rams      beta_Ravens
##       10000.000       10000.000       10950.155       10000.000       10000.000
## beta_Commanders     beta_Saints    beta_Seahawks    beta_Steelers      beta_Titans
##        9616.780       10000.000       10000.000       10000.000       10000.000
##    beta_Vikings
##       10000.000
## [1] "Gelman statistic:"
##       beta_49ers      beta_Bears    beta_Bengals      beta_Bills     beta_Broncos
##       1.0001461       1.0003931       1.0003203       1.0000326       1.0003590
##     beta_Browns  beta_Buccaneers   beta_Cardinals    beta_Chargers      beta_Chiefs
##       1.0000864       1.0003364       0.9999411       1.0000245       1.0001834
##      beta_Colts     beta_Cowboys    beta_Dolphins      beta_Eagles     beta_Falcons
##       0.9998894       1.0001361       1.0001878       1.0000169       0.9999933
##     beta_Giants     beta_Jaguars        beta_Jets       beta_Lions     beta_Packers
##       1.0002092       0.9999380       1.0002135       1.0001726       1.0001756
```

```
##    beta_Panthers    beta_Patriots    beta_Raiders      beta_Rams    beta_Ravens
##        1.0000725        1.0005481        0.9999600      1.0002552      1.0013735
## beta_Commanders      beta_Saints    beta_Seahawks   beta_Steelers     beta_Titans
##        1.0002436        0.9999532        1.0004033      0.9999067      1.0002337
##     beta_Vikings
##        1.0003980
```

Trace Plots of β

Autocorrelation Plots of β
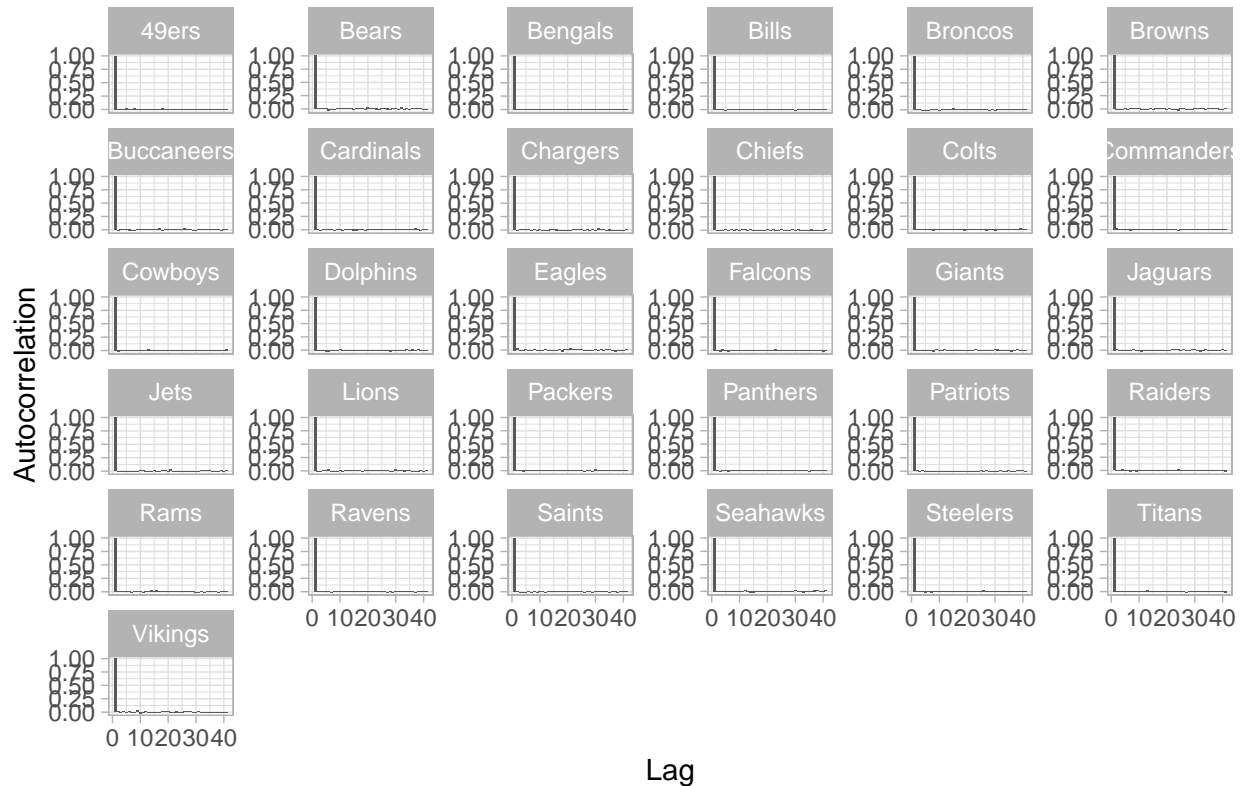

```
## [1] ""
```

Fit looks great. Effective sample size is roughly 10,000 for all parameters. The Gelman statistic is roughly 1.00 for all parameters. Nothing concerning is in the trace plots nor the autocorrelation plots.

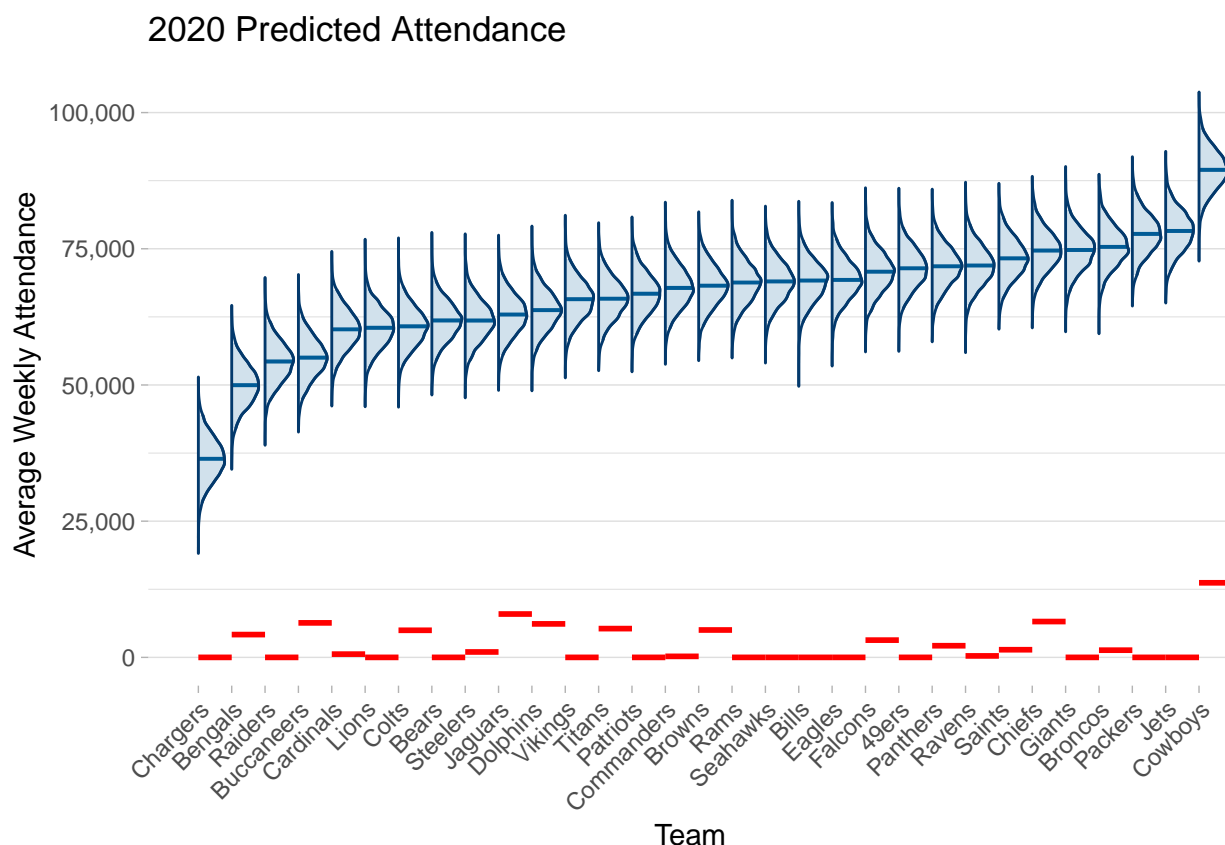## Estimated Revenue Loss

Attendance

Which teams had the biggest attendance loss because of covid?

```
post_predicted_attendance <- get_post_predicted_attendance(
  attendance,
  fit_samps,
  team_names,
  "All teams"
)

plot_attendance(
  post_predicted_attendance,
  attendance_2020,
  team_names,
  "All teams"
)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

2020 Predicted Attendance

The Jets seem to have the worst impact on attendance. The Chargers had the smallest impact, because they had such low attendance in the first place.

Revenue

$$\text{Impact of Covid on Weekly Attendance}_{\text{team}} \approx \text{Actual 2020 Attendance}_{\text{team}} - \text{Est 2020 Attendance}_{\text{team}}$$

$$\text{Est Rev Impact} \approx \sum_{\text{team} \in \text{All teams}} \text{Impact Weekly Attendance}_{\text{team}} \times \text{Avg. 2019 Ticket Cost}_{\text{team}} \times \text{Num Weeks}$$

How much money was actually lost on ticket sales because of covid?

```
post_predicted_loss_attendance <- get_post_predicted_lost_attendance(
  post_predicted_attendance,
  attendance_2020,
  team_names,
  "All teams"
)
print(post_predicted_loss_attendance %>% head())
```

```
##   pred_attendance team_name X          city     Home       diff
## 1        71436.83     49ers 1 San Francisco    0.000 -71436.83
## 2        61808.14     Bears 2       Chicago    0.000 -61808.14
## 3        49956.52   Bengals 3    Cincinnati 4185.312 -45771.21
## 4        69055.67     Bills 4       Buffalo    0.000 -69055.67
## 5        75374.36   Broncos 5        Denver 1320.125 -74054.24
## 6        68229.60    Browns 6     Cleveland 5027.500 -63202.10
```

```r
est_rev_loss <- get_est_lost_rev(
  post_predicted_loss_attendance,
  prices_2019,
  "All teams"
)
```

Estimated Revenue Loss: −$6.13B