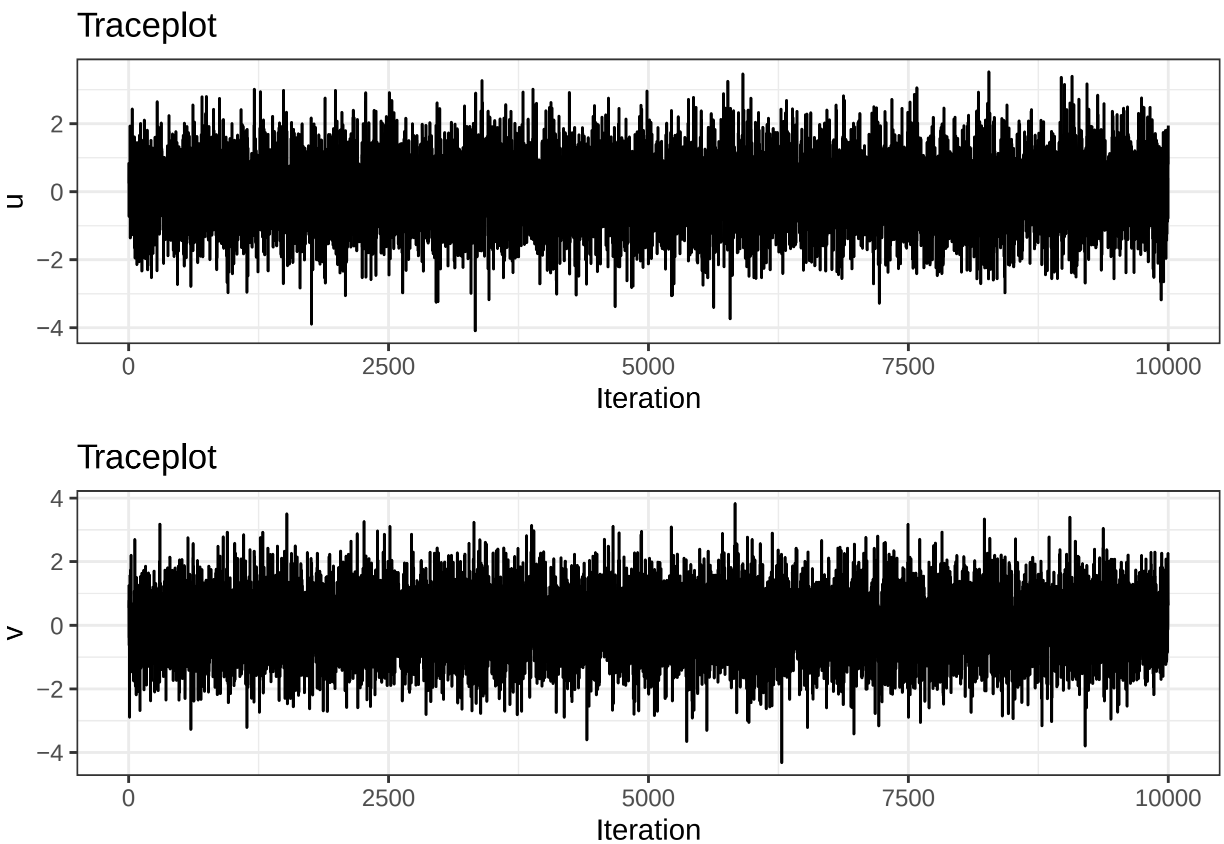
STAT 450: Bayesian Statistics - Homework 5

Problem 1: Gibbs Sampler for Bivariate Normal Distribution

(a)

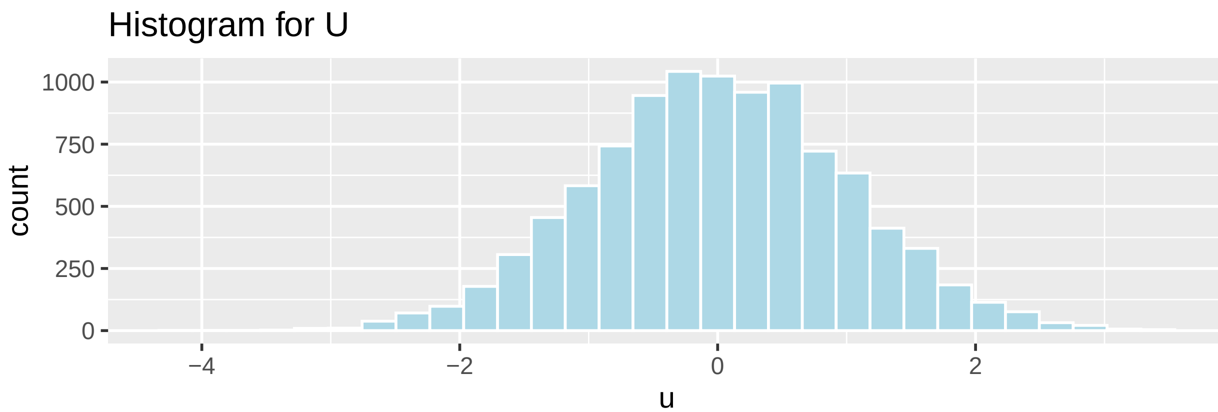
```r
# parameters

p <- 0.3

# number of samples (10000 values of theta, gamma, and ytilde)
nsims <- 10000

#define objects to store these values
Us <- rep(NA, nsims)
Vs <- rep(NA, nsims)
Us[1] <- rnorm(1)
Vs[1] <- rnorm(1)

for(i in 2:nsims){ #iterative process
  Us[i] <- rnorm(1, p*Vs[i-1], sqrt(1-p^2))
  Vs[i] <- rnorm(1, p*Us[i], sqrt(1-p^2))
}


#plots
df <- data.frame(iteration = 1:nsims,
                 u = Us,
                 v = Vs)

h_u <- ggplot(data=df) +
  geom_histogram(mapping = aes(x=u), color = "white", fill = "lightblue") +
  ggtitle("Histogram for U")

h_v <- ggplot(data=df) +
  geom_histogram(mapping = aes(x=v), color = "white", fill = "lightblue") +
  ggtitle("Histogram for V")

# traceplots (line plot of every sample per iteration/ MCMC chain)
tp_u <- ggplot(df, aes(x=iteration, y=u)) + geom_line() +
  labs(x="Iteration", title="Traceplot") +
  theme_bw()

tp_v <- ggplot(df, aes(x=iteration, y=v)) + geom_line() +
  labs(x="Iteration", title="Traceplot") +
  theme_bw()
```

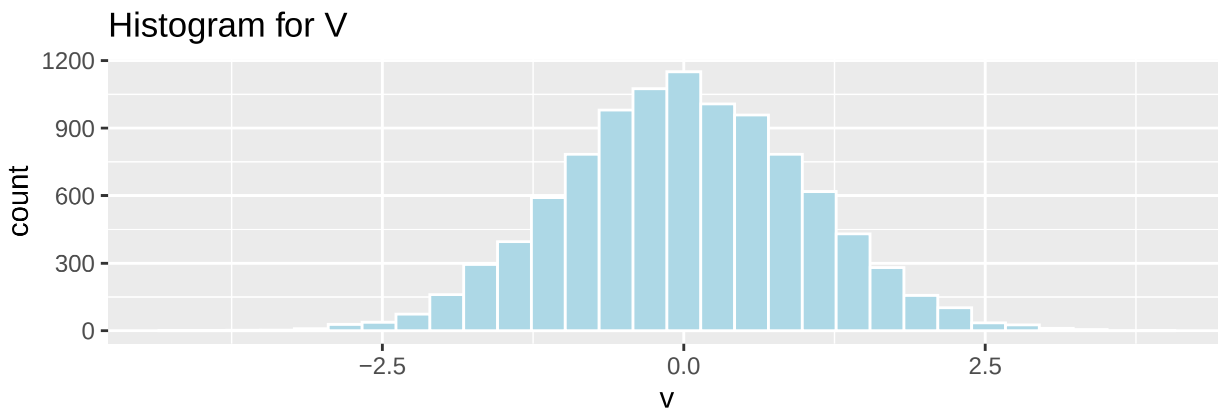
```r
grid.arrange(tp_u, tp_v)
```




```r
grid.arrange(h_u, h_v)
```

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

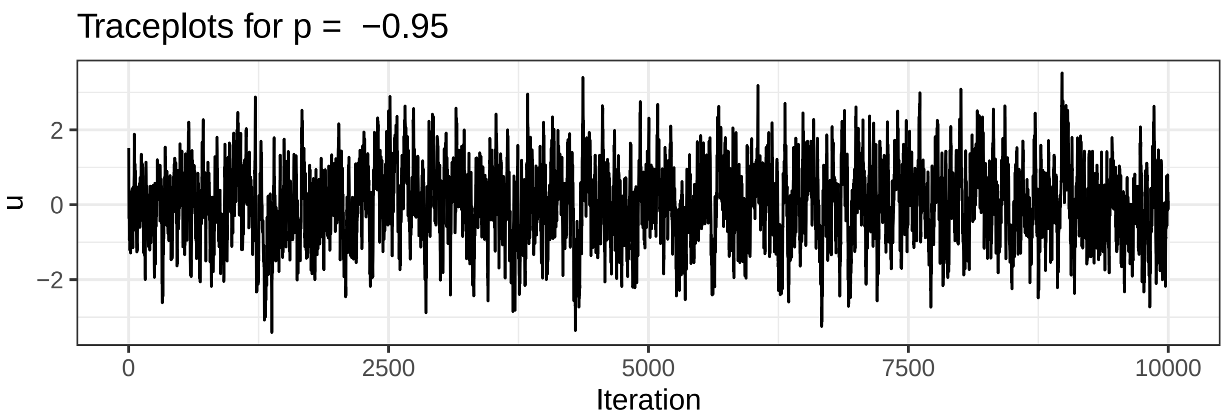


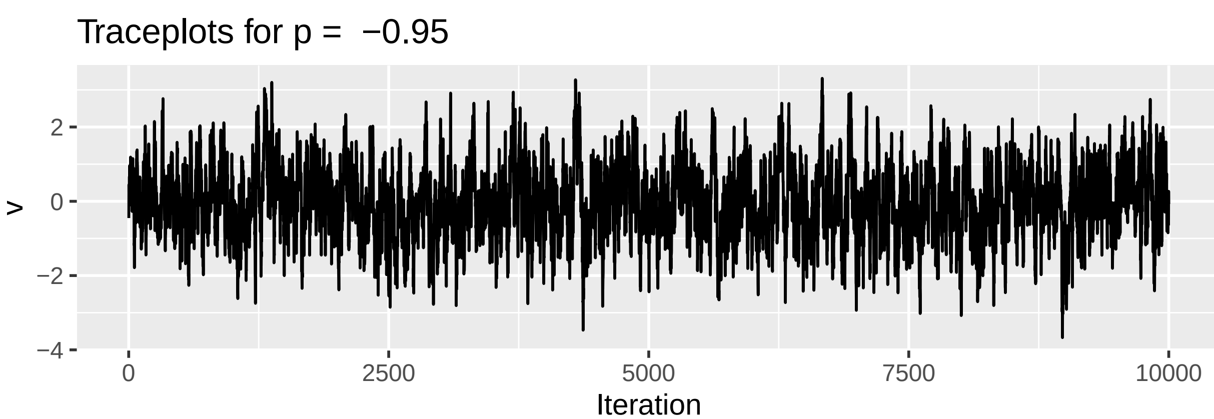


(b)

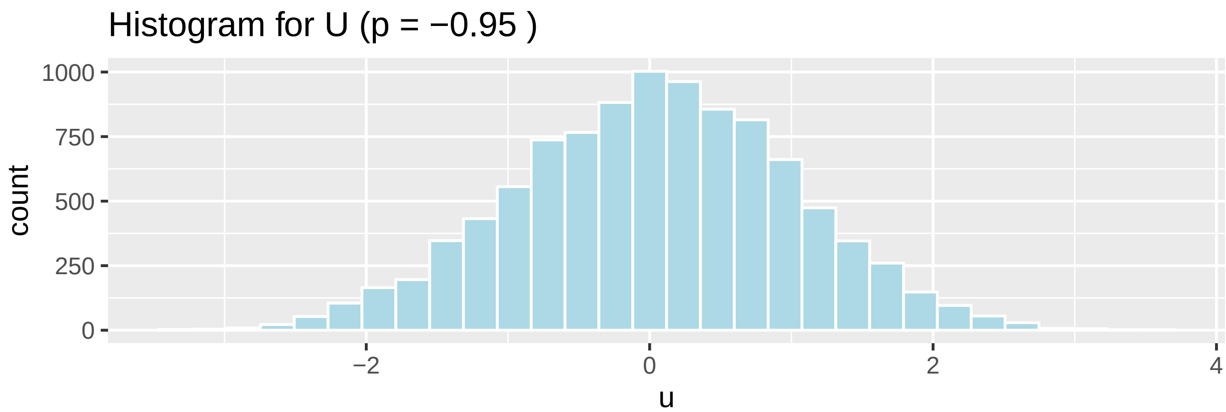
```
set.seed(4505)

Ps <- c(-0.95,0.1,0.95)

Us <- rep(NA, nsims)
Vs <- rep(NA, nsims)
Us[1] <- rnorm(1)
Vs[1] <- rnorm(1)

for (j in 1:3){
  p <- Ps[j]
  for(i in 2:nsims){ #iterative process
    Us[i] <- rnorm(1, p*Vs[i-1], sqrt(1-p^2))
    Vs[i] <- rnorm(1, p*Us[i], sqrt(1-p^2))
  }

  df <- data.frame(iteration = 1:nsims,
                   u = Us,
                   v = Vs)

  # traceplots (line plot of every sample per iteration/ MCMC chain)
  tp_u <- ggplot(df, aes(x=iteration, y=u)) + geom_line() +
    labs(x="Iteration", title = paste('Traceplots for p = ', Ps[j])) +
    theme_bw()
```

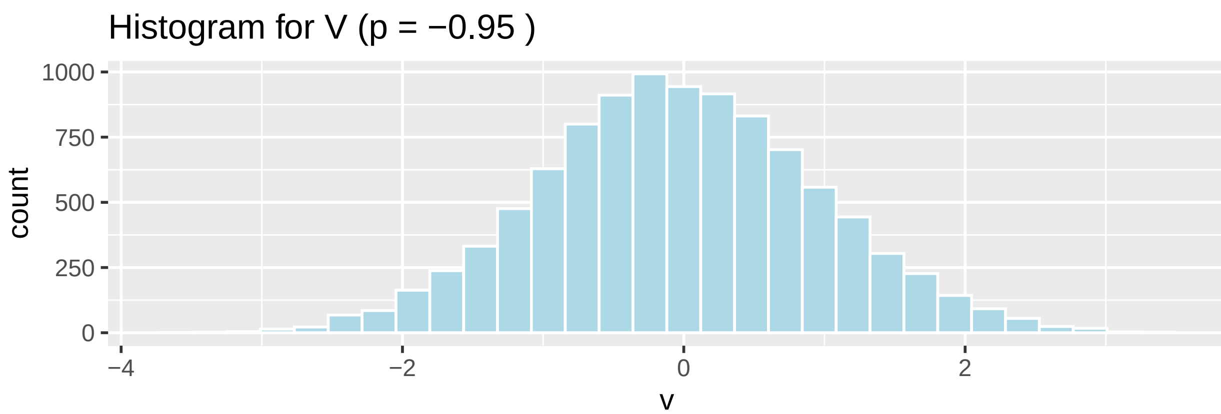
```r
  tp_v <- ggplot(df, aes(x=iteration, y=v)) + geom_line() +
    labs(x="Iteration", title = paste('Traceplots for p = ', Ps[j]))
    theme_bw()

  h_u <- ggplot(df, aes(x = u)) +
    geom_histogram(color = "white", fill = "lightblue") +
    ggtitle(paste("Histogram for U (p =", Ps[j], ")"))

  h_v <- ggplot(df, aes(x = v)) +
    geom_histogram(color = "white", fill = "lightblue") +
    ggtitle(paste("Histogram for V (p =", Ps[j], ")"))

  grid.arrange(tp_u, tp_v)
  grid.arrange(h_u, h_v)

}
```

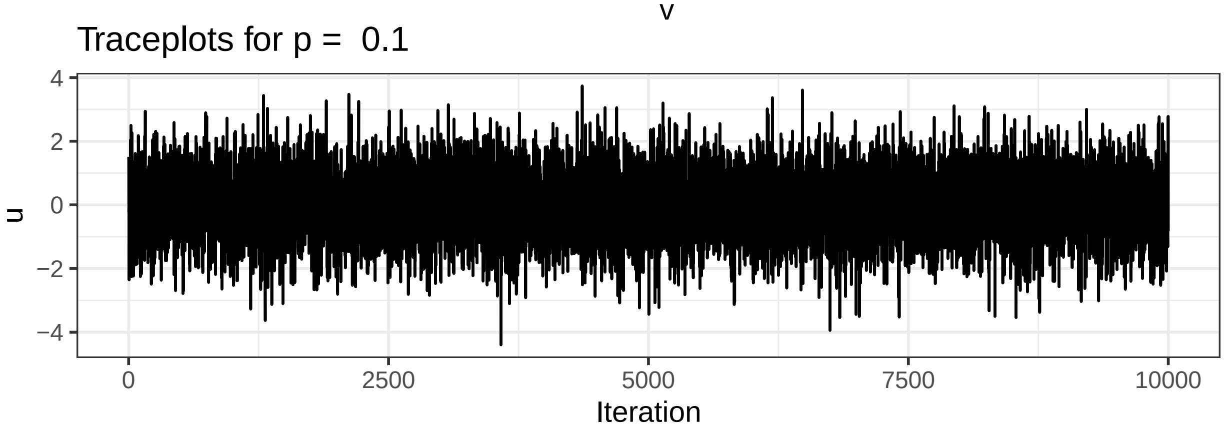


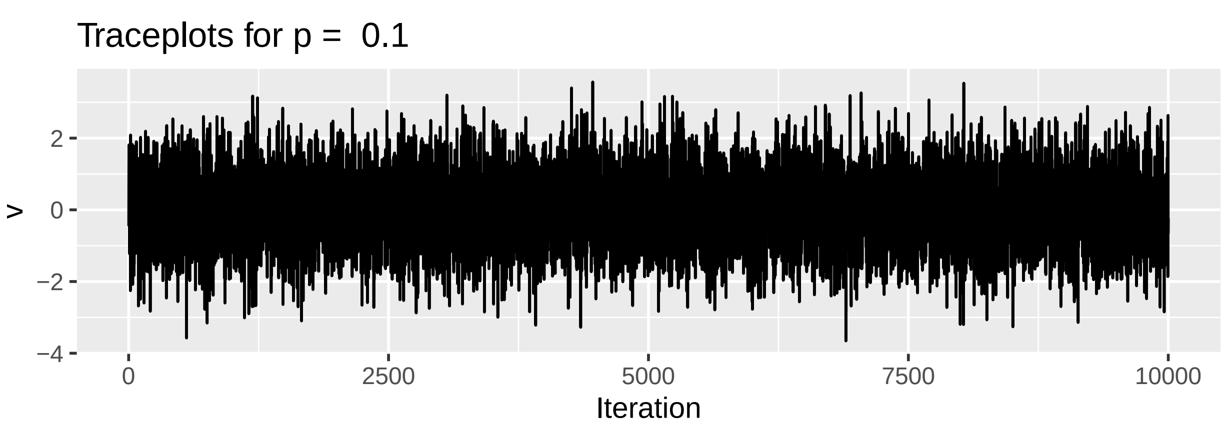



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



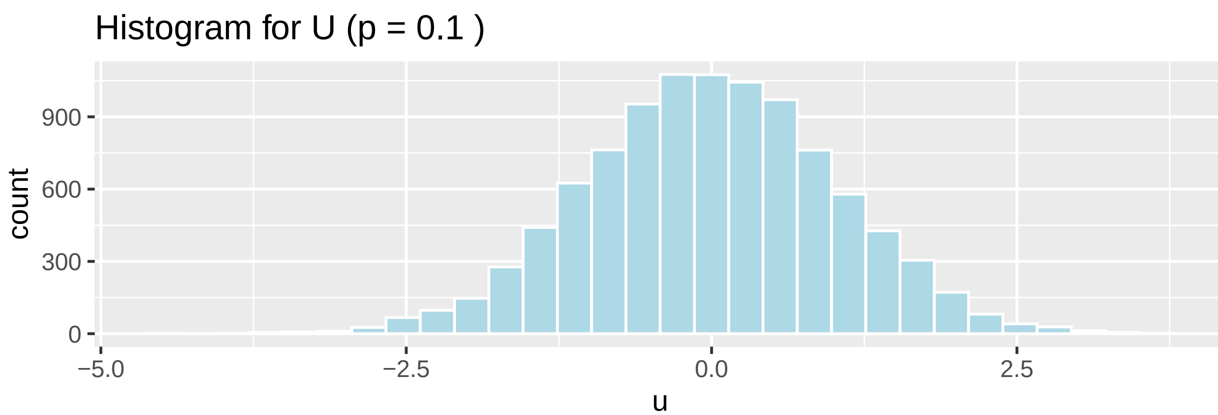


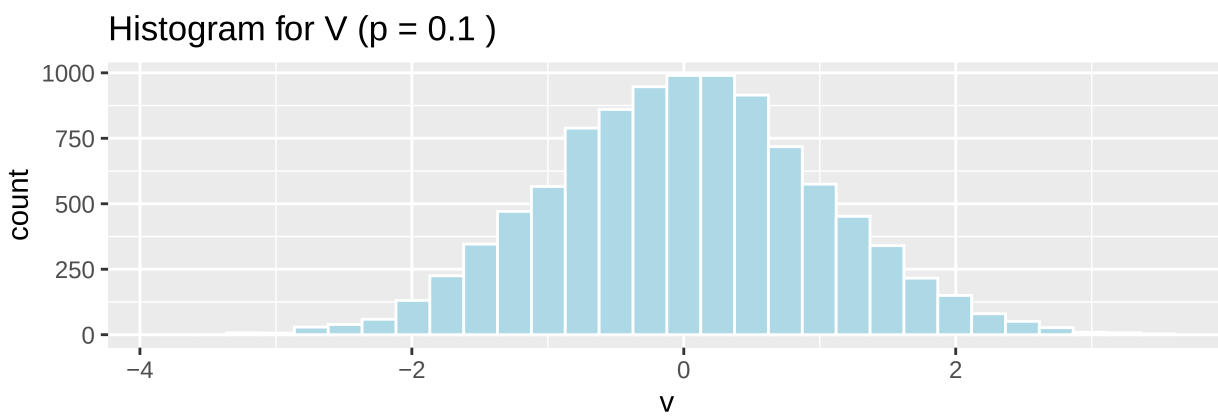


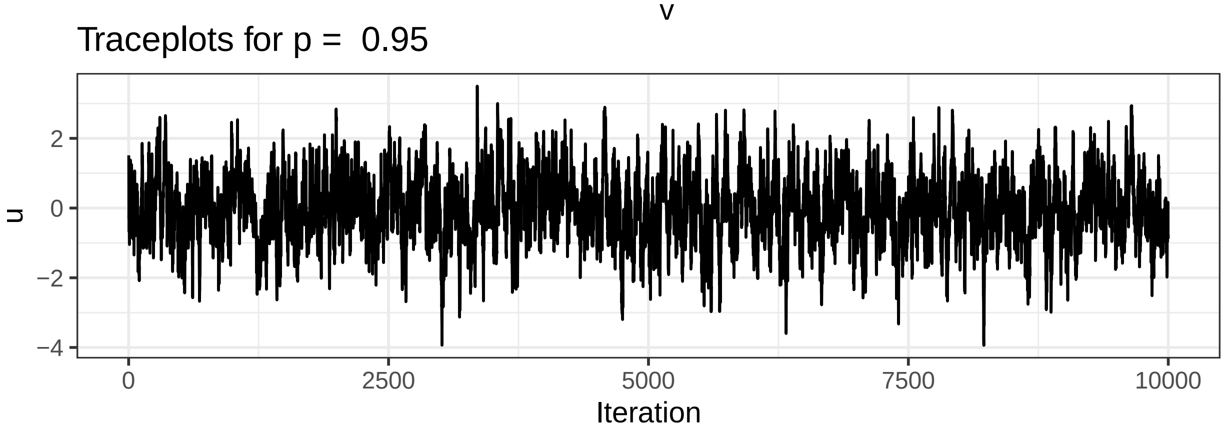


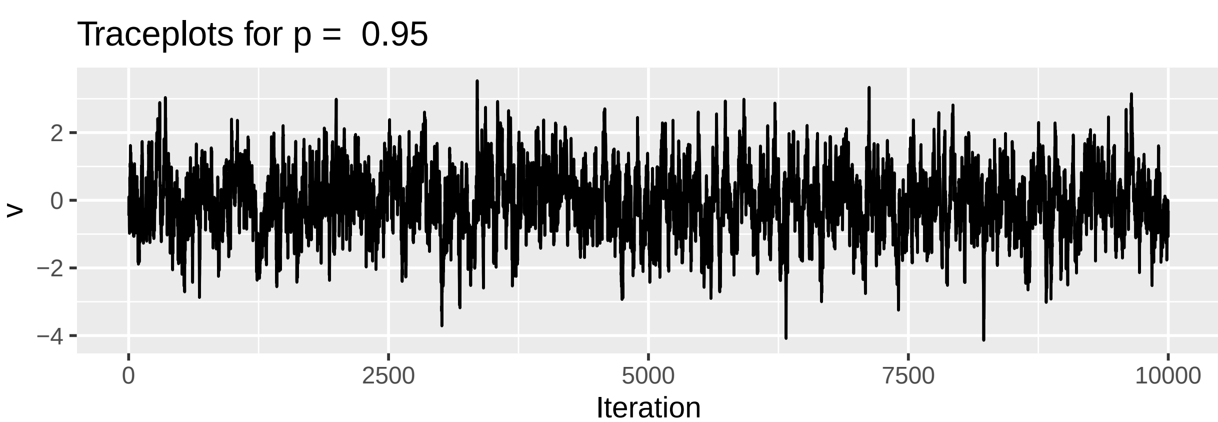

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

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

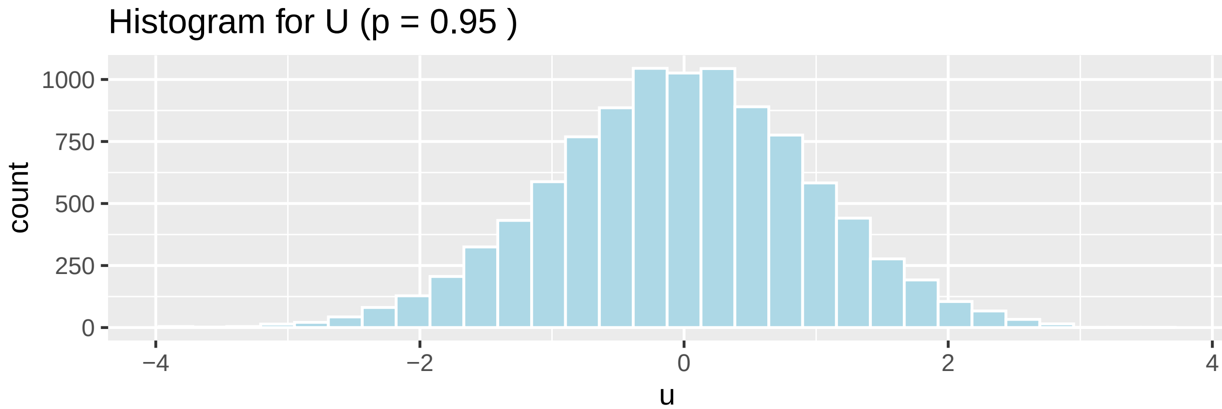


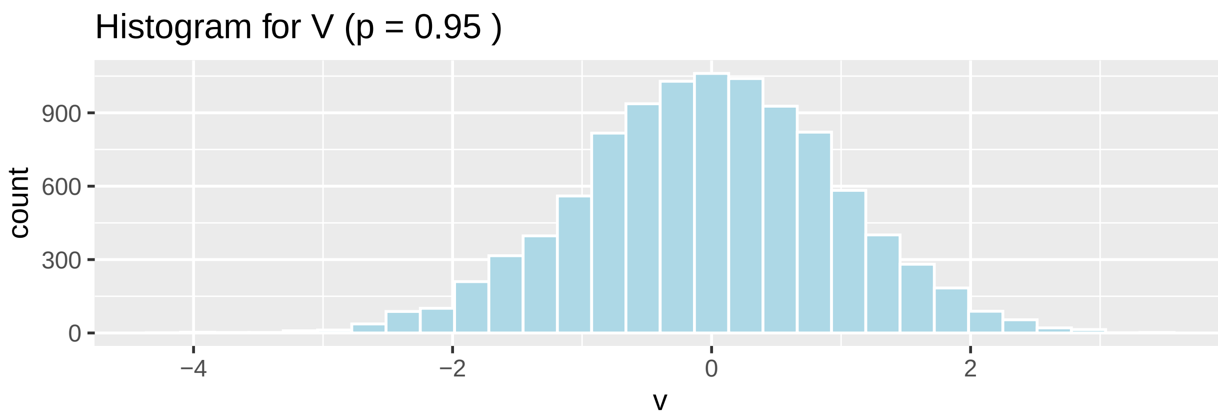






```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





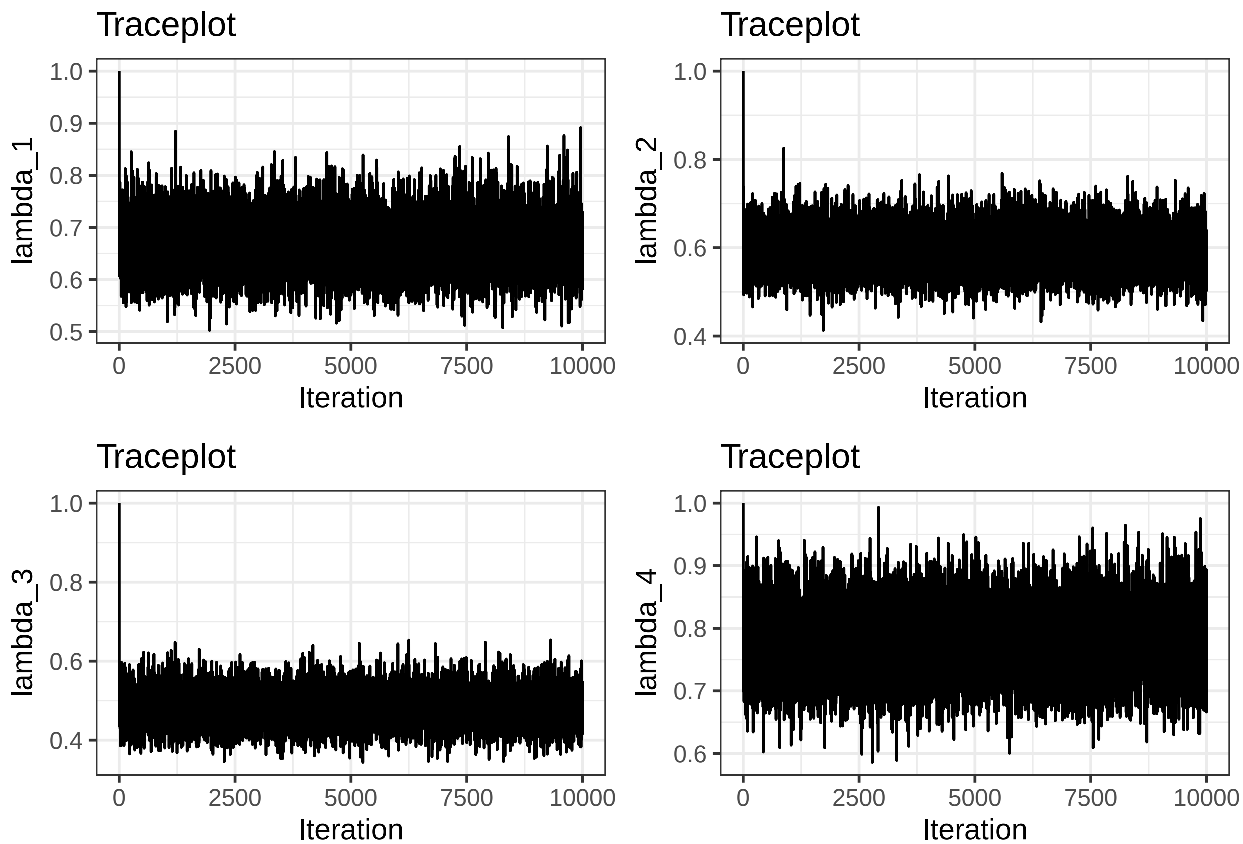
When |p| is near 1, the iterations reach convergence slower.

Problem 2: Gibbs Sampler for NFL Concussions Data

(b)

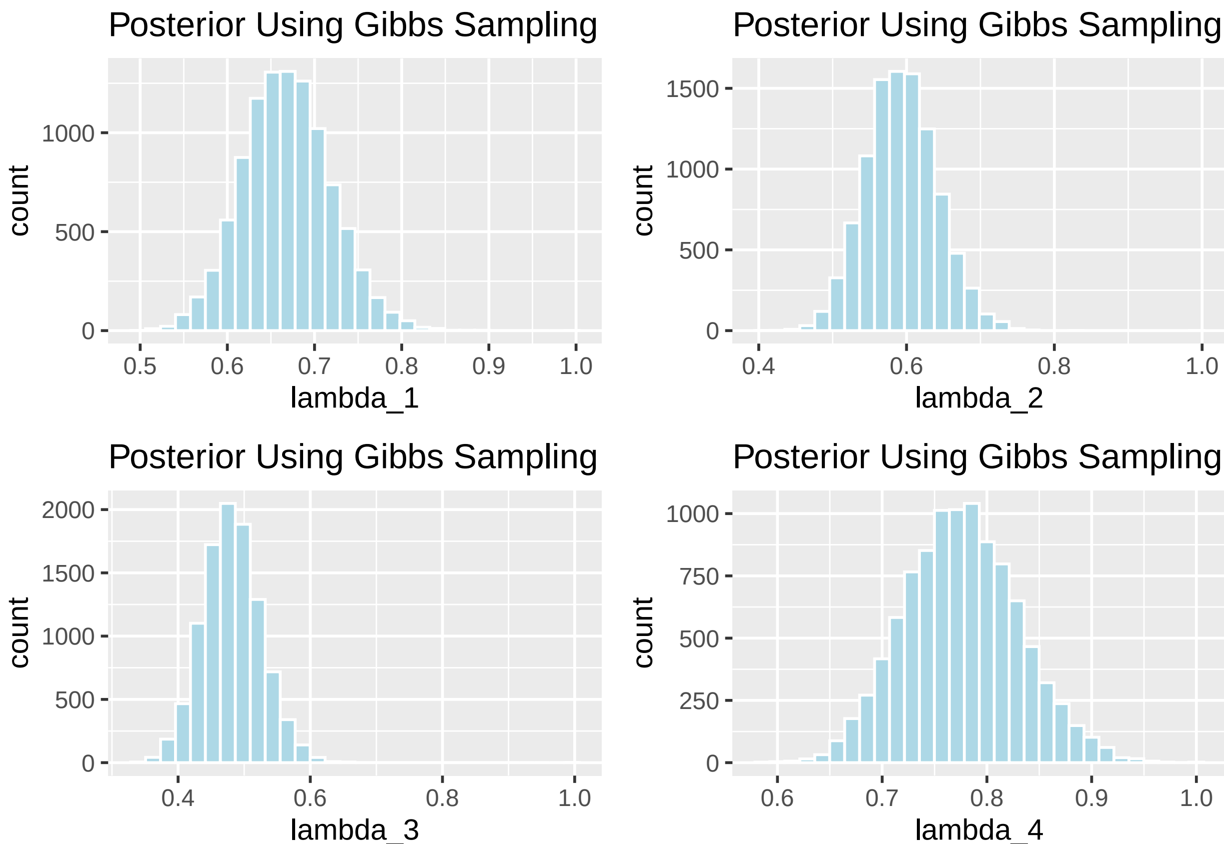
```r
set.seed(4505)

a <- 0.1
b <- 0.1

# data
y <- c(171, 152, 123, 199)
ybar <- mean(y)
n <- 256

# number of samples (number of iterations of the Gibbs samples)
nsims <- 10000

lambdas_1 <- rep(NA, nsims)
lambdas_1[1] <- 1
lambdas_2 <- rep(NA, nsims)
lambdas_2[1] <- 1
```

```
lambdas_3 <- rep(NA, nsims)
lambdas_3[1] <- 1
lambdas_4 <- rep(NA, nsims)
lambdas_4[1] <- 1

gammas <- rep(NA, nsims)
gammas[1] <- 1


for(t in 2:nsims){
  #sample lambda
  lambdas_1[t] <- rgamma(1, y[1] + 1, n + gammas[t-1])
  lambdas_2[t] <- rgamma(1, y[2] + 1, n + gammas[t-1])
  lambdas_3[t] <- rgamma(1, y[3] + 1, n + gammas[t-1])
  lambdas_4[t] <- rgamma(1, y[4] + 1, n + gammas[t-1])

  #sample gamma given all the lambdas
  gammas[t] <- rgamma(1, a + 4, lambdas_1[t] + lambdas_2[t] + lambdas_3[t] + lambdas_4[t] + b)
}

# plots
df <- data.frame(iteration = 1:nsims, lambda_1 = lambdas_1, lambda_2 =lambdas_2, lambda_3 =lambdas_3, la

# traceplots (line plot of every sample per iteration/ MCMC chain)
tp_lambda_1 <- ggplot(df, aes(x=iteration, y=lambda_1)) + geom_line() +
  labs(x="Iteration", y=expression(lambda_1), title="Traceplot") +
  theme_bw()

tp_lambda_2 <- ggplot(df, aes(x=iteration, y=lambda_2)) + geom_line() +
  labs(x="Iteration", y=expression(lambda_2), title="Traceplot") +
  theme_bw()

tp_lambda_3 <- ggplot(df, aes(x=iteration, y=lambda_3)) + geom_line() +
  labs(x="Iteration", y=expression(lambda_3), title="Traceplot") +
  theme_bw()

tp_lambda_4 <- ggplot(df, aes(x=iteration, y=lambda_4)) + geom_line() +
  labs(x="Iteration", y=expression(lambda_4), title="Traceplot") +
  theme_bw()

tp_gamma <- ggplot(df, aes(x=iteration, y=gamma)) + geom_line() +
  labs(x="Iteration", y=expression(gamma), title="Traceplot") +
  theme_bw()

grid.arrange(tp_lambda_1, tp_lambda_2, tp_lambda_3, tp_lambda_4)
```



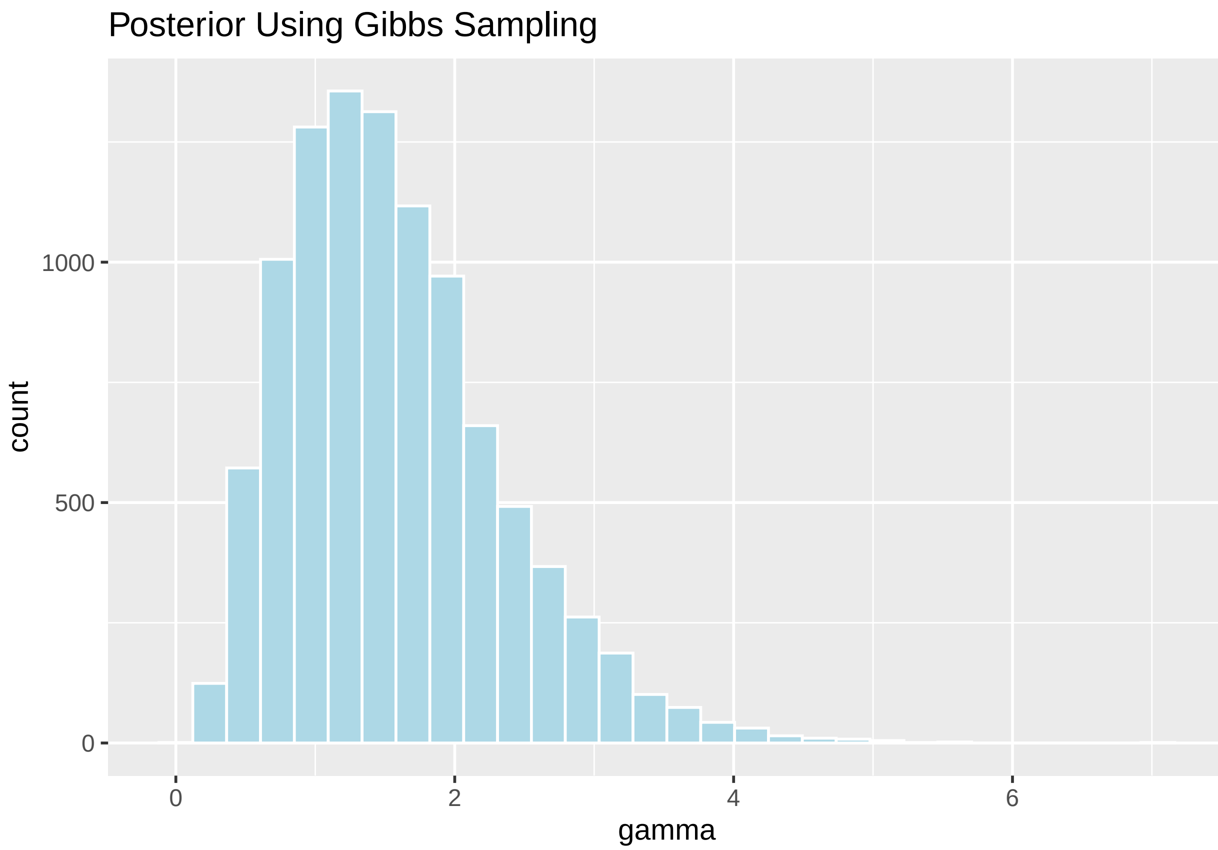

```r
# histograms
hist_lambda_1 <- ggplot(data=df, aes(x=lambda_1)) + geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")

hist_lambda_2 <- ggplot(data=df, aes(x=lambda_2)) + geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")

hist_lambda_3 <- ggplot(data=df, aes(x=lambda_3)) + geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")

hist_lambda_4 <- ggplot(data=df, aes(x=lambda_4)) + geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")


hist_gamma <- ggplot(data=df, aes(x=gamma)) + geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")

grid.arrange(hist_lambda_1, hist_lambda_2, hist_lambda_3, hist_lambda_4)
```

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



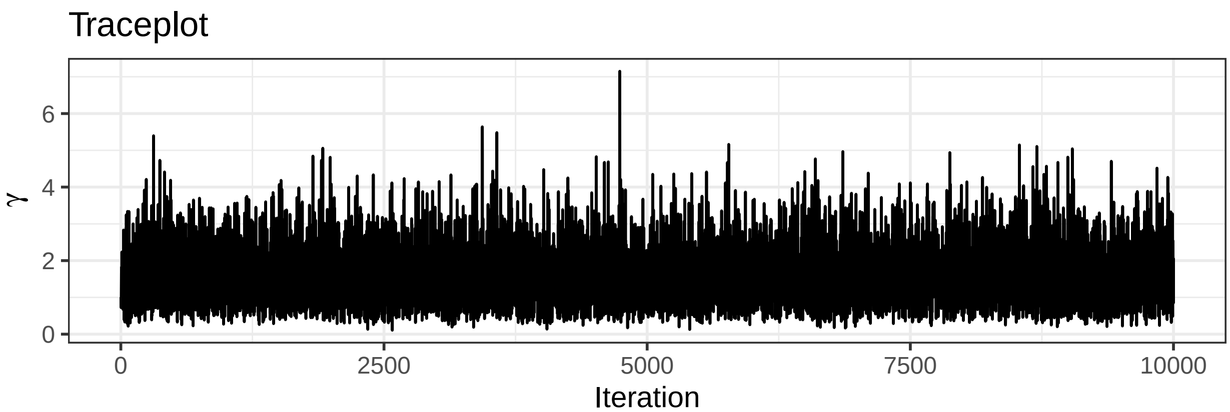

```
hist_gamma
```

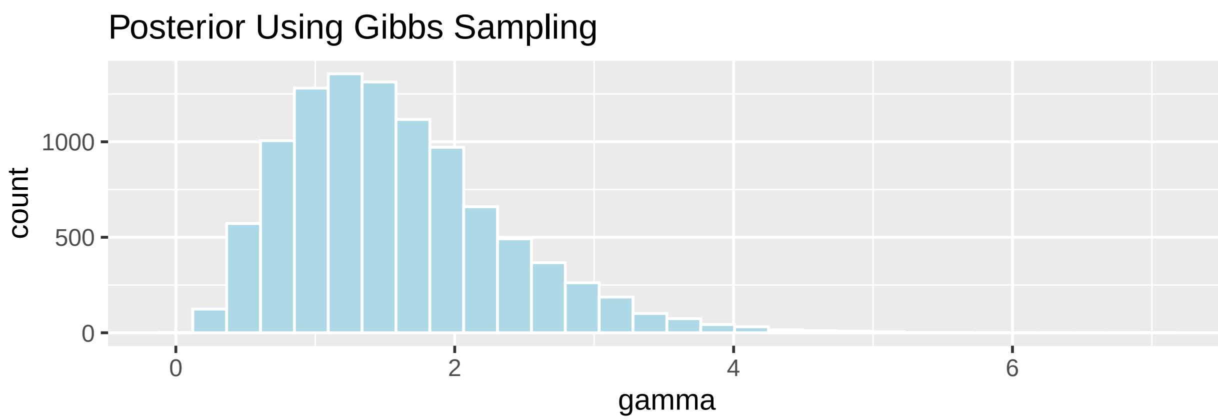
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
grid.arrange(tp_gamma, hist_gamma)
```

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





(c)

```r
set.seed(4505)

# posterior summaries
print("Year: 2012")
```

```
## [1] "Year: 2012"
```

```r
mean(df$lambda_1)
```

```
## [1] 0.6685762
```

```r
median(df$lambda_1)
```

```
## [1] 0.6669532
```

```r
quantile(df$lambda_1, c(0.025, 0.975))
```

```
##      2.5%     97.5% 
## 0.5726969 0.7724731
```

```r
print("Year: 2013")
```

```
## [1] "Year: 2013"
```

```r
mean(df$lambda_2)
```

```
## [1] 0.5933875
```

```r
median(df$lambda_2)
```

```
## [1] 0.5922095
```

```r
quantile(df$lambda_2, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5025651 0.6906371
```

```r
print("Year: 2014")
```

```
## [1] "Year: 2014"
```

```r
mean(df$lambda_3)
```

```
## [1] 0.4815517
```

```r
median(df$lambda_3)
```

```
## [1] 0.4802477
```

```r
quantile(df$lambda_3, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.3969224 0.5720871
```

```r
print("Year: 2015")
```

```
## [1] "Year: 2015"
```

```r
mean(df$lambda_4)
```

```
## [1] 0.7762073
```

```r
median(df$lambda_4)
```

```
## [1] 0.7749239
```

```r
quantile(df$lambda_4, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.6738542 0.8875141
```

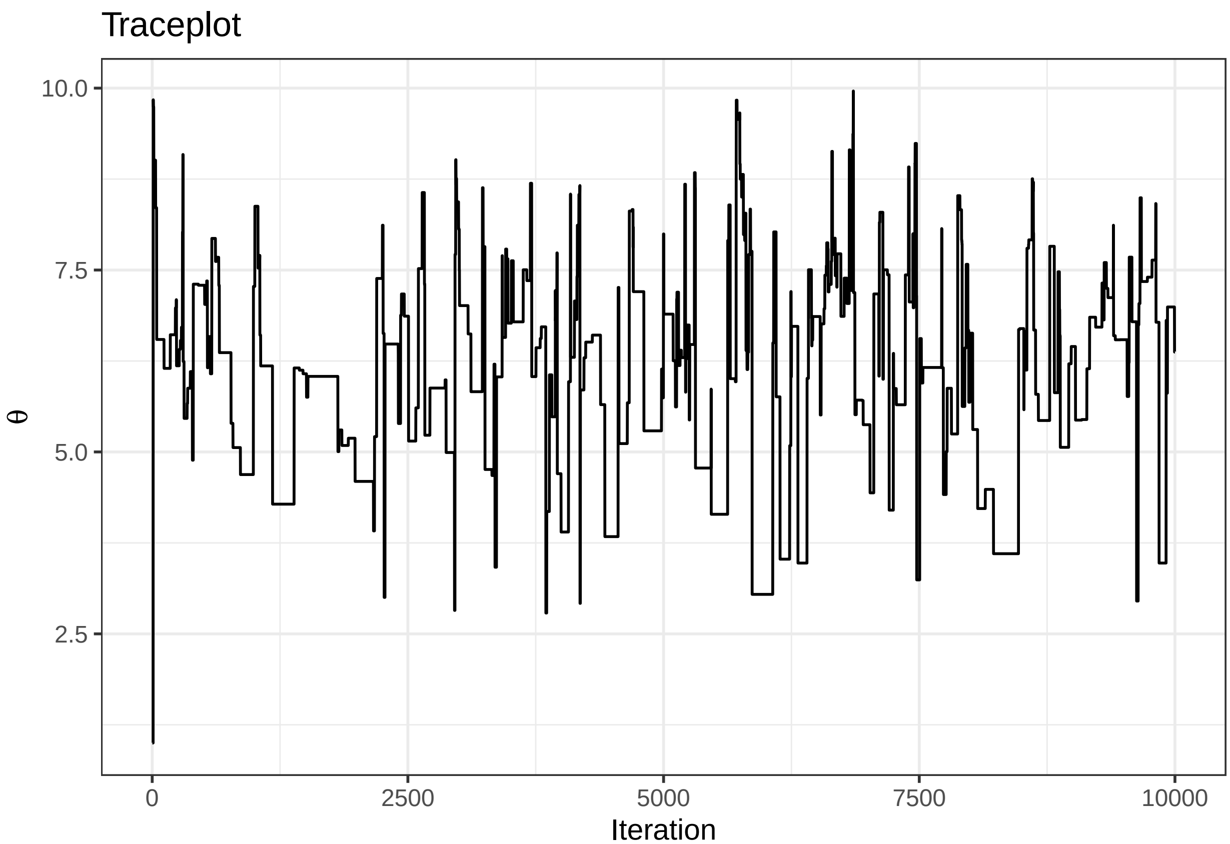
It seems that, on average, the rates of concussions actually decrease from 2012 to 2014, but suddenly saw a sharp rise in 2015.

Problem 3: Metropolis-Hastings Algorithm for ALS Data

(b)

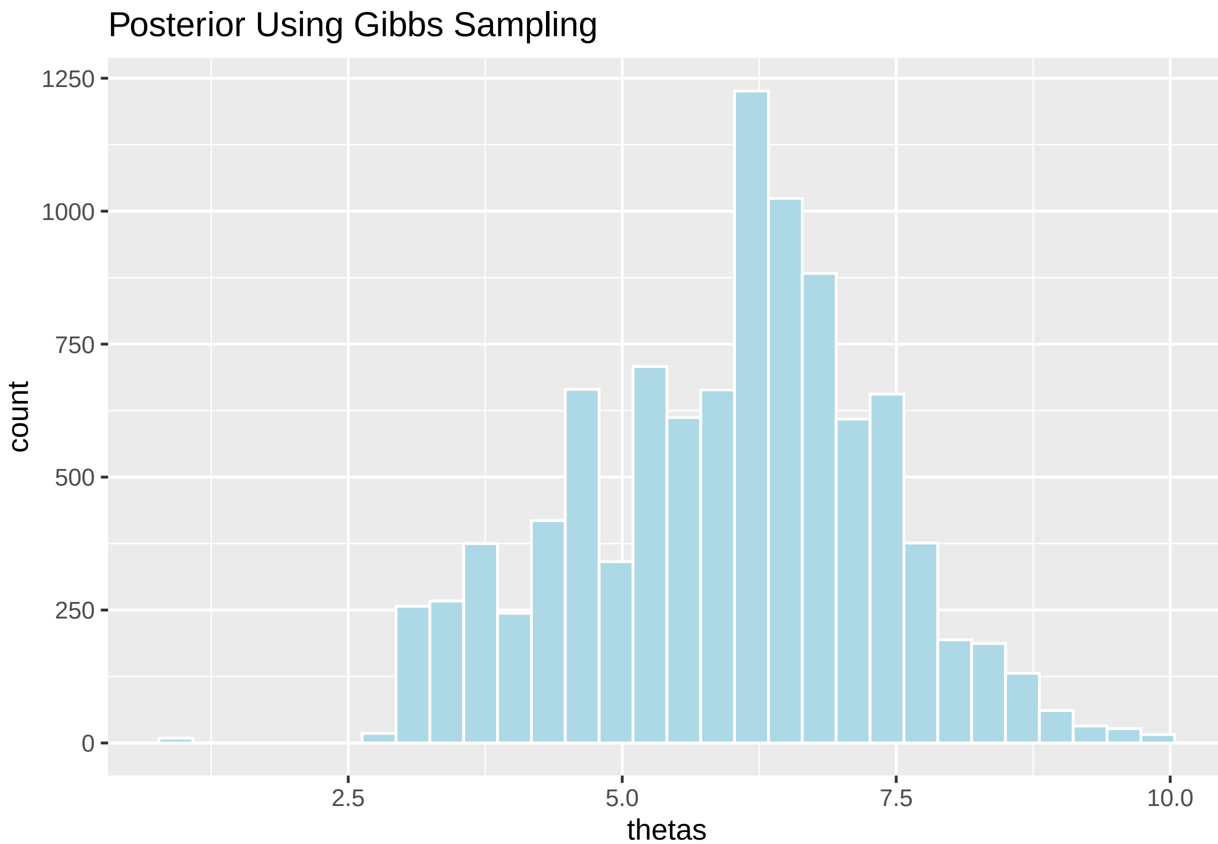
```r
set.seed(4505)
# Independent Metropolis-Hastings - Normal-Cauchy Model

# define densities as functions
p <- function(theta, m1, y1, m2, y2)
{
  return( (theta*m1)^y1 * exp(-(theta*m1)) * (theta*m2)^y2 * exp(-(theta*m2)) * exp(-(1/50) * (theta-2)
}
```

```r
# number of samples
nsims <- 10000

# define objects to store posterior samples
thetas <- rep(NA, nsims)

# specify required values
t <- 1 #first iteration
thetas[t] <- 1 #starting theta value

for(t in 1:(nsims-1)){
  #propose a value
  proposed <- rnorm(1, mean = 13, sd = c(3,1))

  #calculate r ratio, alpha, and Mu
  r <- (p(theta = proposed, 1,5,2,8)/p(theta = thetas[t], 1,5,2,8))
  alpha <- min(1, r) # the acceptance probability
  u <- runif(1, min = 0, max = 1)

  #decision
  thetas[t+1] <- ifelse(u<alpha, proposed, thetas[t])
}

# traceplot
ggplot(data = data.frame(iteration = 1:nsims, thetas), aes(x=iteration, y=thetas)) +
  geom_line() +
  labs(x="Iteration", y=expression(theta), title="Traceplot") +
  theme_bw()
```




```
# histogram
ggplot(data = data.frame(iteration = 1:nsims, thetas), aes(x=thetas)) +
  geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")
```

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




```r
# posterior summaries
mean(thetas)
```

```
## [1] 5.952292
```

```r
quantile(thetas, c(0.025, 0.975))
```

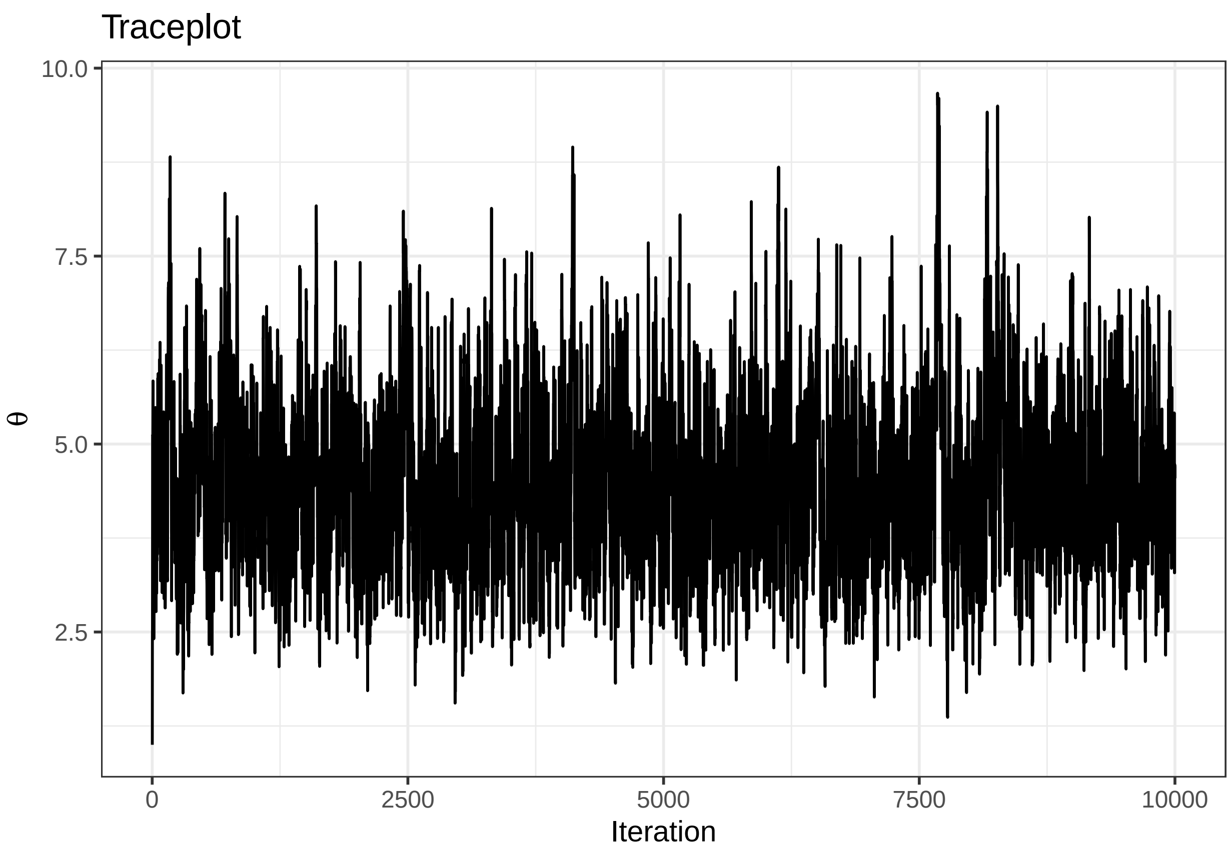
```
##     2.5%    97.5%
## 3.042836 8.522050
```

I did the posterior distribution wrong, so r and alpha always come out to be extremly tiny. But I don't know where I went wrong.

(c)

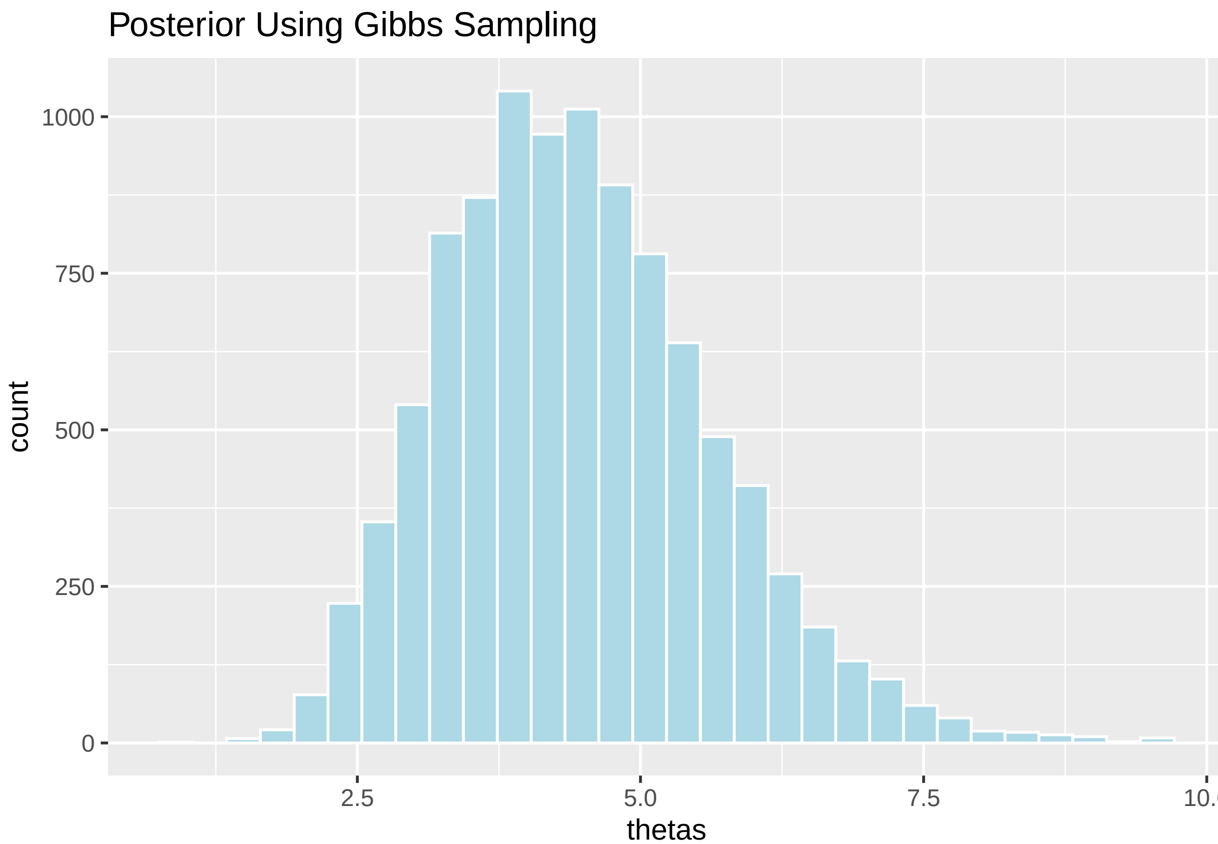
```r
set.seed(4505)
# full sampler to generate posterior samples

thetas[t] <- 13/3 #starting theta value

for(t in 1:(nsims-1)){
  #propose a value
  proposed <- rnorm(1, mean = thetas[t], sd =1) #proposed changes from mean = y to mean = current

  #calculate r ratio, alpha, and Mu
  r <- (p(theta = proposed, 1,5,2,8)/p(theta = thetas[t], 1,5,2,8))
  alpha <- min(1, r) # the acceptance probability
  u <- runif(1, min = 0, max = 1) #Uniform(0,1) random variable
```

```r
  #decision
  thetas[t+1] <- ifelse(u<alpha, proposed, thetas[t])
}


# traceplot
ggplot(data = data.frame(iteration = 1:nsims, thetas), aes(x=iteration, y=thetas)) +
  geom_line() +
  labs(x="Iteration", y=expression(theta), title="Traceplot") +
  theme_bw()
```




```r
# histogram
ggplot(data = data.frame(iteration = 1:nsims, thetas), aes(x=thetas)) +
  geom_histogram(color="white", fill="lightblue") +
  ggtitle("Posterior Using Gibbs Sampling")
```

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




```r
# posterior summaries
summary(thetas)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.607   4.354   4.455   5.185   9.667
```

```r
quantile(thetas, c(0.025, 0.975))
```

```
##     2.5%    97.5%
## 2.438140 7.091661
```