

A Bayesian Approach to Salary Analysis for Data Scientists

Vera Bruzzese

2025-10-01

Contents

1. Introduction	2
2. Exploratory analysis	2
2.1 Import and Cleaning	2
2.2 Our variables	3
2.3 Visualisation	4
Salary and log-salary distribution	4
Log-Salary by Experience Level	6
Log-Salary by Company Size	6
Log-Salary by Remote Ratio	7
Log-Salary Distribution in Top 10 Countries	8
2.4 Correlation and ANOVA Analyses	9
3. Mixed-Effects Model	10
4. Nimble Model	11
4.1 Setting everything up	11
4.2 Running the MCMC Sampler	13
4.3 Summary table	14
4.4 Trace plots	15
5. PPC	16
6. Conclusion	18

1. Introduction

The role of the data scientist has become one of the most sought-after professions in the global labor market in recent years. As organizations of all sizes and across all industries have come to rely on data-driven decision-making, there has been a corresponding growth in demand for professionals whose expertise spans data science, machine learning, and data analysis. This high demand has brought questions about salary distribution, geographic disparities, and the influence of professional characteristics like experience level, company size, and remote work arrangements to the forefront of discussions about this workforce.

While traditional descriptive analyses, like summary statistics and regression models, offer a useful starting point for examining salary structures in this field, they have their limitations. The Bayesian approach allows for a better quantification of uncertainty and enhances the interpretability of parameter estimates using prior information.

In this analysis, we use a global dataset of data science salaries to investigate the factors that influence compensation. Our process begins with an examination of the raw and log-transformed salary distributions, which is then followed by an exploratory comparison across different experience levels, company sizes, remote work arrangements, and geographic locations. Drawing from these initial insights, we then develop a Bayesian model using NIMBLE. This model is designed to account for the fixed effects of experience level and company size, while also modeling company location variations as a random effect. This methodology allows us to estimate the relative impact of experience and company size on salaries, while also taking into account the systematic differences that exist between countries.

2. Exploratory analysis

2.1 Import and Cleaning

```
data <- read.csv("C:/Users/gbruz/OneDrive/Desktop/applied/ds_salaries.csv")

data <- unique(data) %>%
  select(salary_in_usd, experience_level, remote_ratio, company_location, company_size) %>%
  filter(!is.na(salary_in_usd)) %>%
  mutate(
    log_salary = log(salary_in_usd),
    experience_level = factor(experience_level, levels = c("EN", "MI", "SE", "EX")),
    company_size = as.factor(company_size),
    company_location = as.factor(company_location),
    remote_ratio = as.numeric(remote_ratio)
  )

head(data)
```

```
## salary_in_usd experience_level remote_ratio company_location company_size
## 1      85847             SE           100             ES           L
## 2      30000             MI           100             US           S
## 3      25500             MI           100             US           S
## 4      175000            SE           100             CA           M
## 5      120000            SE           100             CA           M
## 6      222200            SE            0             US           L
## log_salary
## 1    11.36032
## 2    10.30895
## 3    10.14643
## 4    12.07254
## 5    11.69525
## 6    12.31133
```

The dataset `ds_salaries.csv` was loaded and cleaned before starting the analysis. I first removed duplicate rows and kept only the columns relevant for the study (salary, experience level, remote ratio, company size, and company location). Any rows with missing salaries were excluded.

I also created a new variable, **log_salary**, the natural log of the salary, which should make the distribution less skewed.

Experience level, company size, and company location were turned into factors, while the remote ratio was kept as a numeric variable (0, 50, 100).

2.2 Our variables

Let's summarize the main variables used in the analysis:

- `salary_in_usd`: numerical variable representing the annual salary in USD.
- `log_salary`: numerical variable, the natural logarithm of `salary_in_usd`, used to stabilize variance and reduce skewness.
- `experience_level`: categorical variable indicating the level of experience of the employee (EN, MI, SE, EX).
- `company_size`: categorical variable describing the size of the company (S, M, L).
- `company_location`: categorical variable indicating the country where the company is located.

```
summary(data$salary_in_usd)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5132   84975  130000  133409  175000  450000
```

```
summary(data$log_salary)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      8.543  11.350  11.775   11.635  12.073   13.017
```

```
table(data$experience_level)
```

```
##
##      EN      MI      SE      EX
##      270    664   1554     96
```

```
table(data$company_size)
```

```
##
##      L      M      S
##      409   2028   147
```

```
table(data$company_location)
```

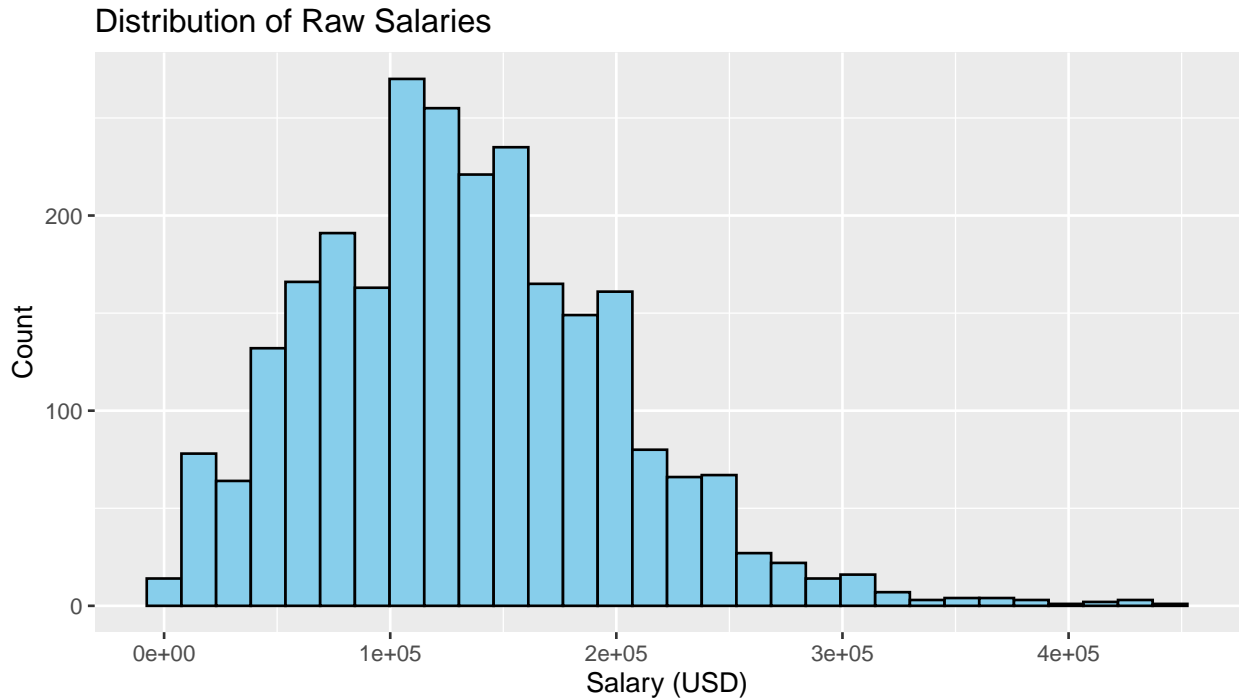
```
##
##      AE      AL      AM      AR      AS      AT      AU      BA      BE      BO      BR      BS      CA      CF      CH      CL
##      3       1       1       3       3       6      14       1       4       1      15       1      83       2       5       1
##      CN      CO      CR      CZ      DE      DK      DZ      EE      EG      ES      FI      FR      GB      GH      GR      HK
##      1       4       1       3      55       4       1       2       1      44       3      33     155       2      11       1
##      HN      HR      HU      ID      IE      IL      IN      IQ      IR      IT      JP      KE      LT      LU      LV      MA
##      1       3       2       2       7       2      57       1       1       4       6       2       2       3       4       1
##      MD      MK      MT      MX      MY      NG      NL      NZ      PH      PK      PL      PR      PT      RO      RU      SE
##      1       1       1      10       1       5      13       1       1       4       5       4      14       2       3       2
##      SG      SI      SK      TH      TR      UA      US      VN
##      6       4       1       3       5       4     1929       1
```

2.3 Visualisation

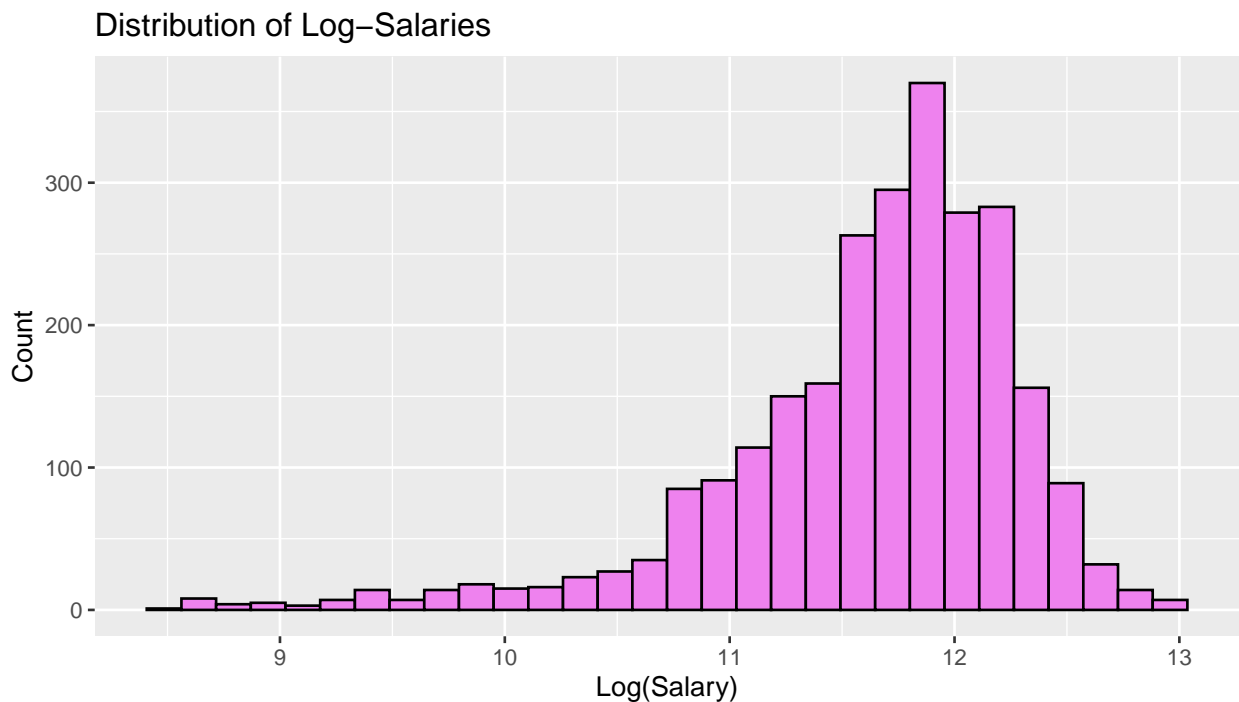
Salary and log-salary distribution

We can visualise the distribution of salaries using histograms.

```
ggplot(data, aes(x = salary_in_usd)) +
  geom_histogram(fill = "skyblue", color = "black") +
  labs(title = "Distribution of Raw Salaries", x = "Salary (USD)", y = "Count")
```



```
ggplot(data, aes(x = log_salary)) +  
  geom_histogram(fill = "violet", color = "black") +  
  labs(title = "Distribution of Log-Salaries", x = "Log(Salary)", y = "Count")
```

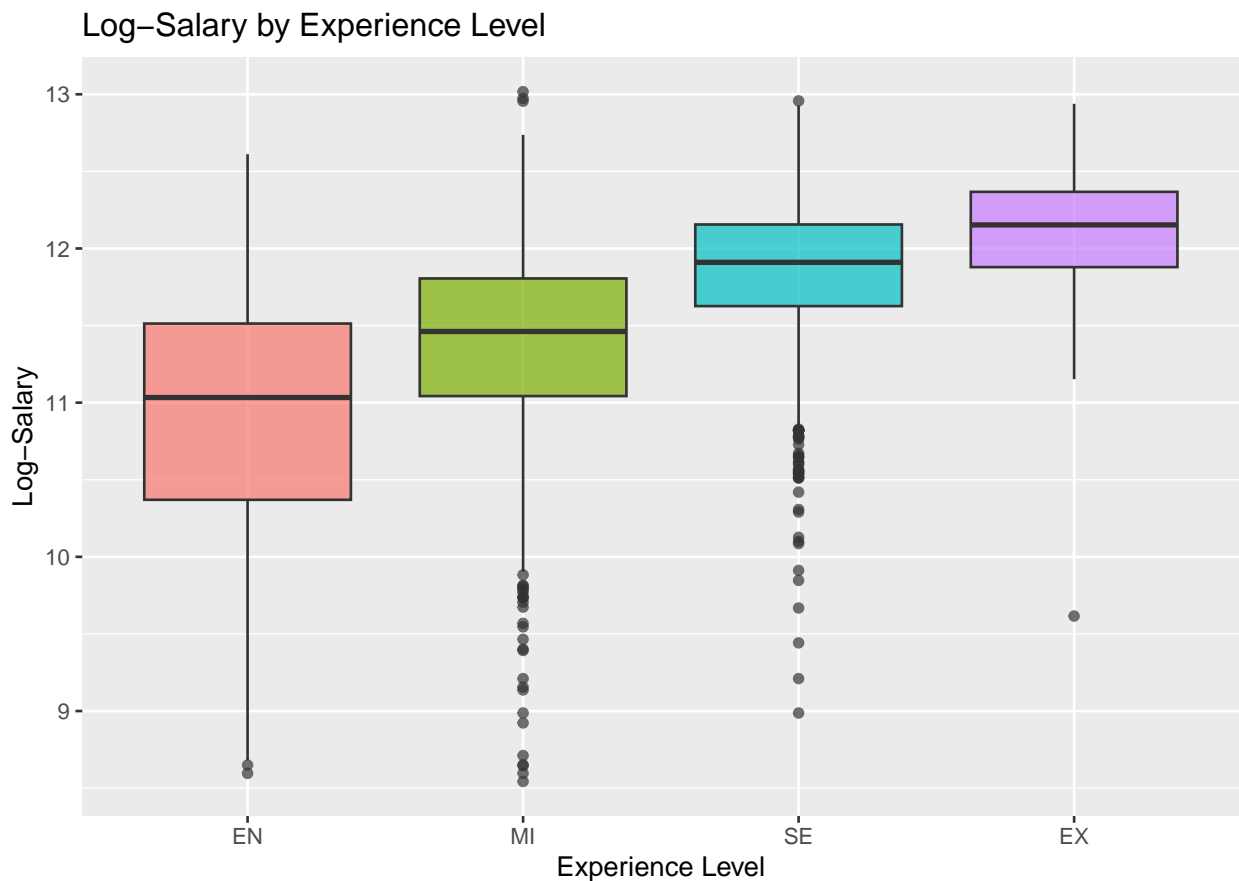


The first plot shows the raw annual salaries (`salary_in_usd`), which are highly right-skewed, indicating a few very high salaries relative to the majority of the data.

The second plot shows the natural logarithm of salaries (`log_salary`), which reduces the right skew and produces a more symmetric distribution.

Log-Salary by Experience Level

```
ggplot(data, aes(x = experience_level, y = log_salary, fill = experience_level)) +  
  geom_boxplot(alpha=0.7) +  
  labs(title = "Log-Salary by Experience Level",  
       x = "Experience Level", y = "Log-Salary") +  
  theme(legend.position = "none")
```

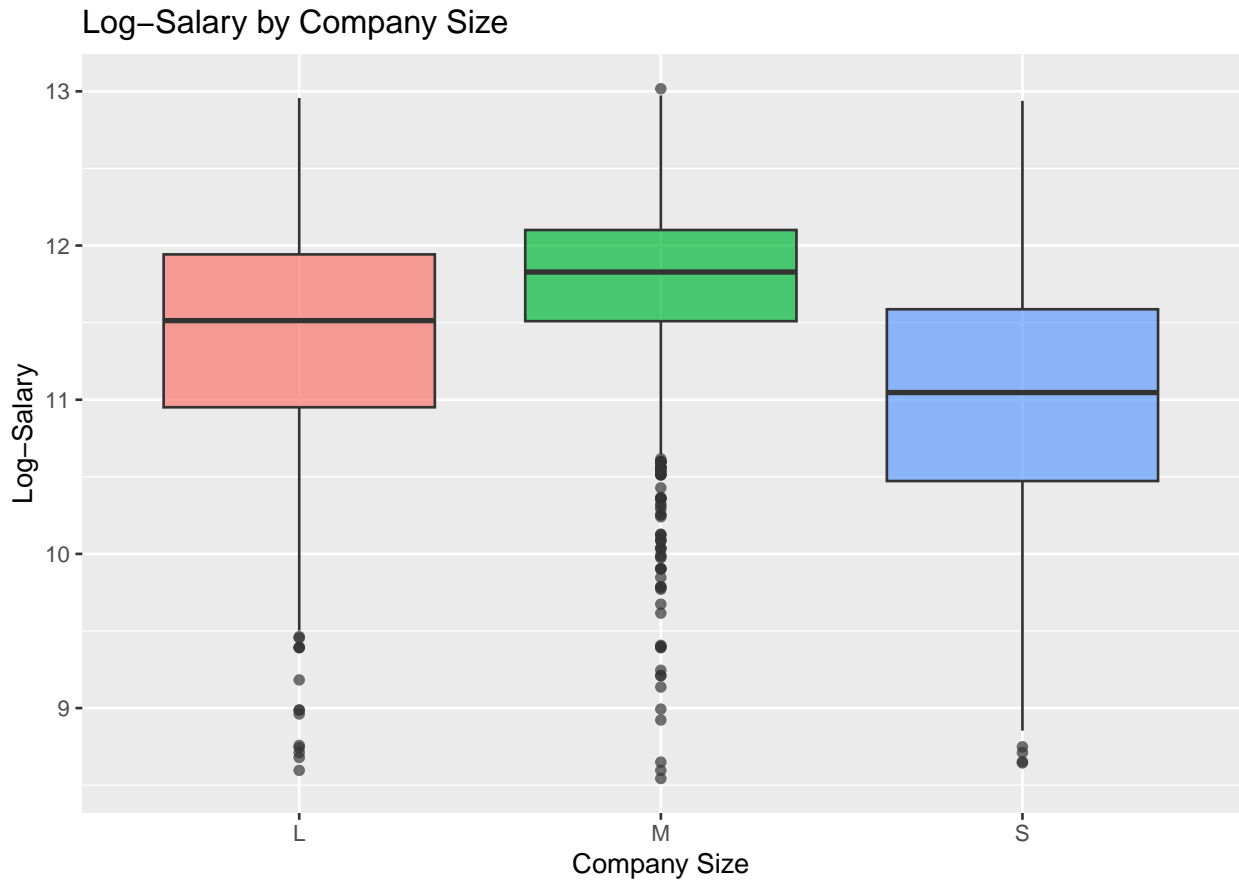


The boxplot shows the distribution of `log_salary` across different experience levels (Entry-level, Mid-level, Senior-level, Executive-level). Salaries grow consistently with experience: entry-level workers earn the least, while executives have the highest pay.

Log-Salary by Company Size

```
ggplot(data, aes(x = company_size, y = log_salary, fill = company_size)) +  
  geom_boxplot(alpha = 0.7) +
```

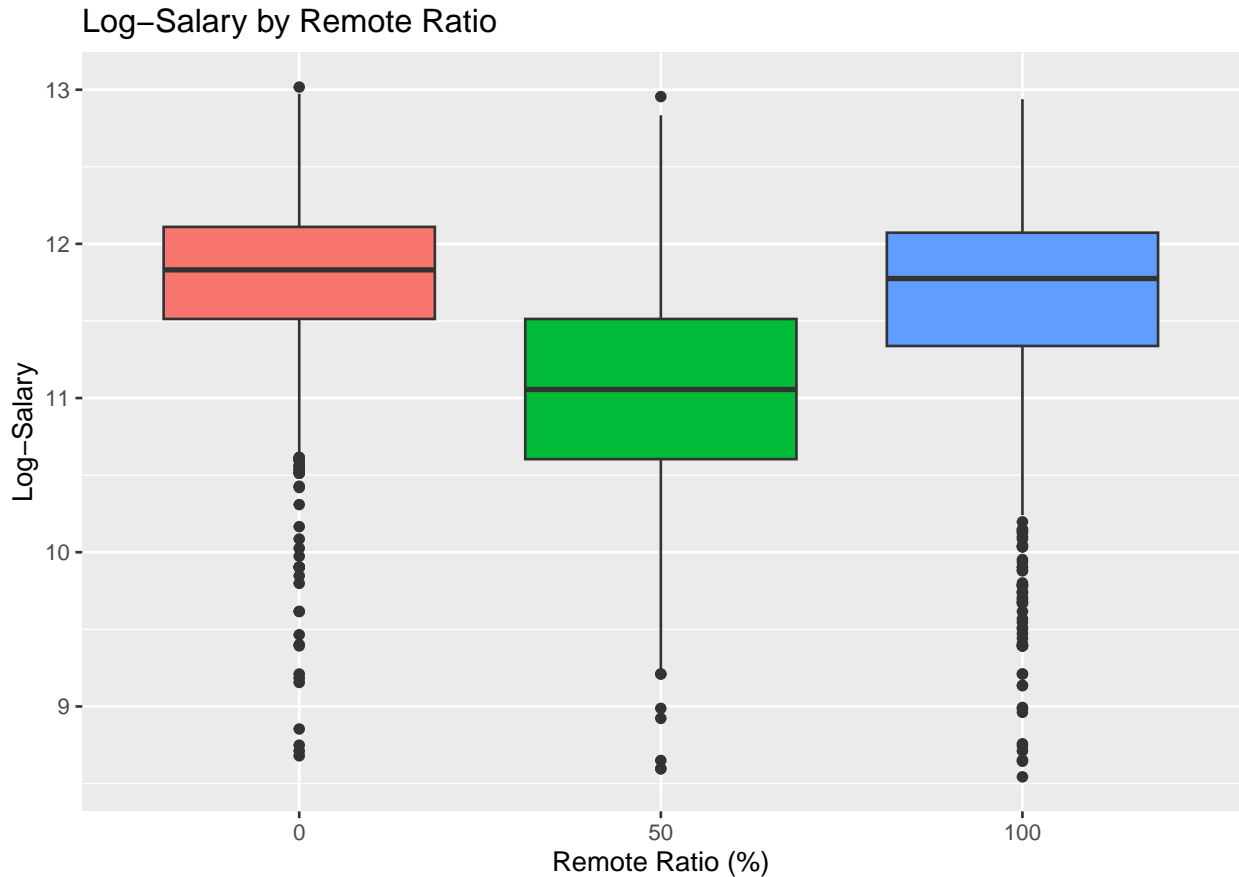
```
labs(title = "Log-Salary by Company Size",
     x = "Company Size", y = "Log-Salary") +
theme(legend.position = "none")
```



The boxplot illustrates the distribution of log-transformed salaries (`log_salary`) across company sizes (S, M, L).

Log-Salary by Remote Ratio

```
ggplot(data, aes(x = factor(remote_ratio), y = log_salary, fill = factor(remote_ratio))) +
  geom_boxplot() +
  labs(title = "Log-Salary by Remote Ratio",
       x = "Remote Ratio (%)", y = "Log-Salary") +
  theme(legend.position = "none")
```

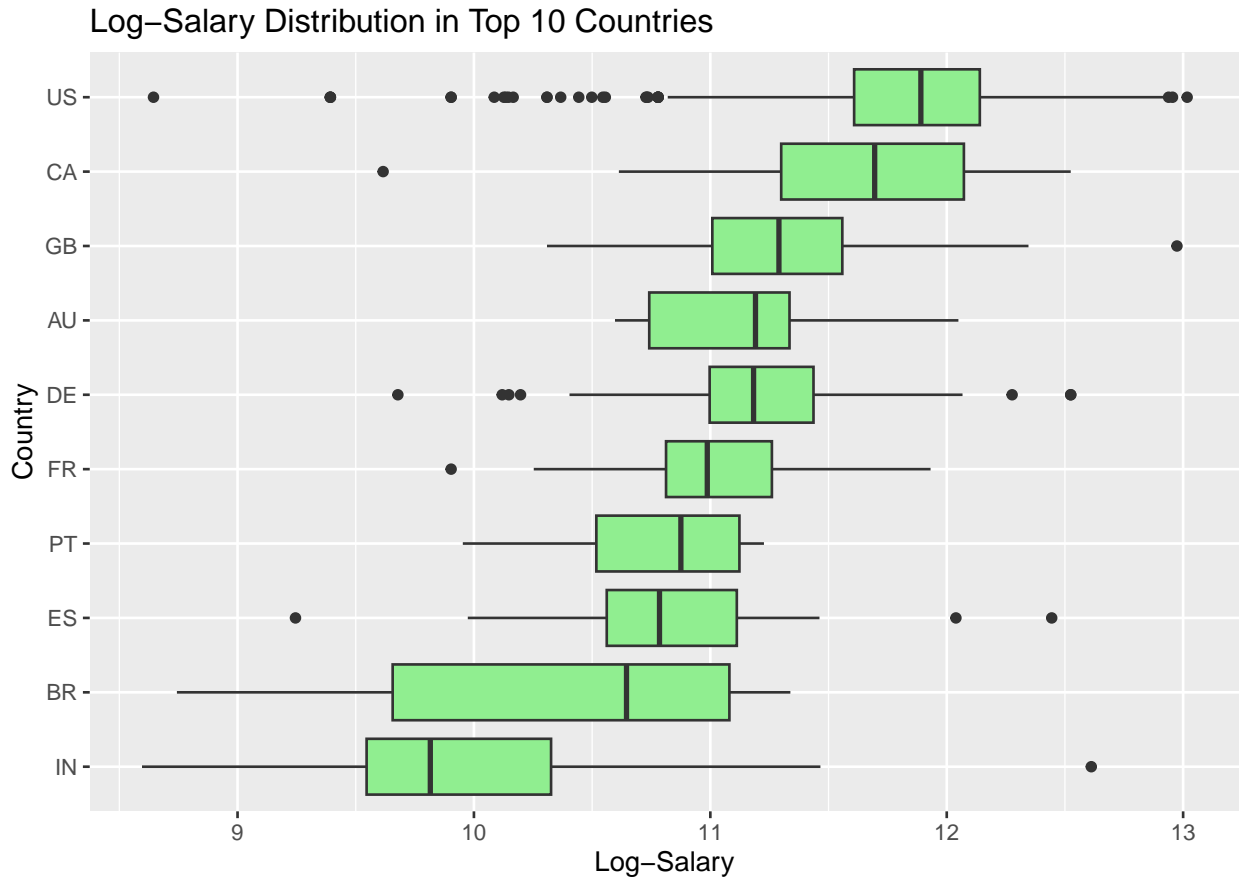


There's some variation depending on how much remote work is allowed, but the effect is weaker compared to experience or company size.

Log-Salary Distribution in Top 10 Countries

```
top_countries <- data %>%
  count(company_location, sort = TRUE) %>%
  slice_head(n = 10) %>%
  pull(company_location)

ggplot(data %>% filter(company_location %in% top_countries),
       aes(x = reorder(company_location, log_salary, median), y = log_salary)) +
  geom_boxplot(fill = "lightgreen") +
  coord_flip() +
  labs(title = "Log-Salary Distribution in Top 10 Countries",
       x = "Country", y = "Log-Salary")
```



Median salaries vary significantly between countries. Median salaries can vary a lot, showing that geography is a big factor at play.

2.4 Correlation and ANOVA Analyses

```
cor_remote <- cor(data$remote_ratio, data$log_salary, use = "complete.obs")
print(cor_remote)
```

```
## [1] -0.09803317
```

The correlation between remote ratio and salary is weakly negative (around -0.10). In other words, people working more remotely tend to earn slightly less, but the effect is small.

```
anova_experience <- aov(log_salary ~ experience_level, data = data)
summary_experience <- summary(anova_experience)

anova_size <- aov(log_salary ~ company_size, data = data)
summary_size <- summary(anova_size)
```

```
anova_location <- aov(log_salary ~ company_location, data = data)
summary_location <- summary(anova_location)

print(summary_experience)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## experience_level    3  295.9   98.65    310 <2e-16 ***
## Residuals          2580  821.0    0.32
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(summary_size)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## company_size      2  122.4   61.18   158.8 <2e-16 ***
## Residuals         2581  994.6    0.39
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(summary_location)
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## company_location   71  564.6    7.952   36.16 <2e-16 ***
## Residuals          2512  552.4    0.220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analysis of Variance (ANOVA) is a statistical test needed for categorical–continuous relationships, used to check whether the means of several groups differ significantly. It does this by comparing the variation between group means to the variation within groups. If the between-group differences are much larger, we conclude that the factor has an effect.

The ANOVA tests show that: **Experience level** has the strongest effect: salaries clearly rise with more seniority. **Company size** also plays a role, though less strongly. **Country of employment** explains a lot of the variation, confirming that geography is a major driver of salaries.

3. Mixed-Effects Model

Our exploratory data analysis revealed that `experience_level` and `company_size` have a clear and predictable influence on salaries. As such, these variables are treated as fixed effects, meaning we are interested in estimating the specific, constant effect of each level (e.g., the salary difference between an entry-level and a senior-level data scientist).

On the other hand, the `company_location` variable presents a different challenge. With a large number of countries represented in the dataset, treating location as a fixed effect would introduce a high number of parameters into our model. This would not only make the model unnecessarily complex but also potentially lead to overfitting, especially for countries with only a few data points.

To address this, we model `company_location` as a random effect. This approach assumes that the salary variations across different countries are not entirely independent but are drawn from a common underlying distribution. Instead of estimating the specific effect of each country, we estimate the parameters of this distribution—specifically, its variance. This allows us to account for the country-level variability in a more parsimonious and robust manner. By treating `company_location` as a random effect, our model can “borrow strength” from the data across all countries to make more stable and reliable estimates, particularly for locations with sparse data.

This mixed-effects approach provides a more flexible and realistic representation of the data, leading to more generalizable insights into the factors that shape data science compensation.

4. Nimble Model

4.1 Setting everything up

To implement our model, we use the Nimble package. We define several constants: `N` for the total number of observations, and `J_exp`, `K_size`, and `L_loc` for the number of unique levels in each categorical variable

```
experience_level_int <- as.integer(data$experience_level)
company_size_int <- as.integer(data$company_size)
company_location_int <- as.integer(data$company_location)

N <- nrow(data)
J_exp <- nlevels(data$experience_level)
K_size <- nlevels(data$company_size)
L_loc <- nlevels(data$company_location)
y <- data$log_salary

modelCode <- nimbleCode({
  for (i in 1:N) {
    log_salary[i] ~ dnorm(mu[i], sd = sigma)
    mu[i] <- alpha +
      beta_exp[experience_level_int[i]] +
      gamma_size[company_size_int[i]] +
      delta_loc[company_location_int[i]]
  }

  alpha ~ dnorm(mu_alpha, sd = 5)
  sigma ~ T(dnorm(0, sd = 1), 0, )
})
```

```

beta_exp[1] <- 0
for (j in 2:J_exp) {
  beta_exp[j] ~ dnorm(0, sd = 2)
}

gamma_size[1] <- 0
for (k in 2:K_size) {
  gamma_size[k] ~ dnorm(0, sd = 2)
}

for (l in 1:L_loc) {
  delta_loc_raw[l] ~ dnorm(0, 1)
  delta_loc[l] <- delta_loc_raw[l] * sigma_loc
}
sigma_loc ~ T(dnorm(0, sd = 1), 0, )
})

```

Likelihood: For each observation i , the `log_salary` variable is modeled as a normal distribution. The expected value of this distribution is a linear combination of the effects we aim to estimate. The residual standard deviation `sigma` captures the variability not explained by the model.

The linear Component ($\mu[i]$): The expected log-salary is composed of:

`alpha`: a global intercept representing the baseline log-salary.

`beta_exp`: the coefficients for experience levels, treated as fixed effects. The first level (Entry-level) is fixed to zero to serve as a baseline for comparison.

`gamma_size`: the coefficients for company sizes, also treated as fixed effects. The first level is fixed to zero as a baseline.

`delta_loc`: the specific offsets for each company location, modeled as random effects.

Priors: Prior distributions are defined for all parameters to be estimated:

The priors for the intercept (`alpha`) and the fixed-effects coefficients (`beta_exp`, `gamma_size`) are weakly informative normal distributions.

The priors for the standard deviation parameters (`sigma` and `sigma_loc`) are half-normal distributions to ensure their sampled values are always positive.

The random effect `delta_loc` is defined hierarchically: each country-specific effect is drawn from a common normal distribution with a mean of zero and a standard deviation of `sigma_loc`. It is this `sigma_loc` parameter that estimates the overall salary variability across different countries. The non-centered parameterization separates the standard deviation (`sigma_loc`) from the raw draws, which improves the efficiency of the MCMC sampler and leads to more stable convergence, especially when the group-level variance is small or weakly identified.

4.2 Running the MCMC Sampler

Once the model is defined, the next step is to fit it to the data by drawing samples from the posterior distribution using a Markov Chain Monte Carlo (MCMC) algorithm. This is handled by the `nimbleMCMC` function.

Before running the sampler, we organize the necessary components:

constants: A list containing fixed values that do not change during the MCMC run, such as the number of observations (N) and the integer-coded categorical variables.

data: A list containing the observed data, in this case, the `log_salary` vector `y`.

initial_values: A function that generates random starting points for each chain.

We then set the MCMC sampler in motion using the `nimbleMCMC` function. To ensure our results are reliable, we don't run the simulation just once; instead, we run three independent Markov chains simultaneously. The core idea is that if all three chains, starting from different random points, end up describing the same answer, we can be much more confident in our findings.

Each of these chains takes a long walk, generating a total of 50,000 samples for the parameters we're monitoring. We don't keep the first 10,000 samples, this initial phase is called "burn-in" period, and it gives the algorithm time to warm up and move from its random starting position to the high-probability region of the posterior distribution.

```
constants <- list(
  N = N, J_exp = J_exp, K_size = K_size, L_loc = L_loc,
  experience_level_int = experience_level_int,
  company_size_int = company_size_int,
  company_location_int = company_location_int,
  mu_alpha = mean(y)
)

nimble_data <- list(log_salary = y)

initial_values <- function() {
  list(
    alpha = rnorm(1, mean(y), 0.5),
    sigma = runif(1, 0.1, 1),
    beta_exp = c(NA, rnorm(J_exp - 1, 0, 0.1)),
    gamma_size = c(NA, rnorm(K_size - 1, 0, 0.1)),
    delta_loc_raw = rnorm(L_loc, 0, 0.1),
    sigma_loc = runif(1, 0.01, 1)
  )
}

set.seed(456)
mcmc_output <- nimbleMCMC(
  code = modelCode,
  constants = constants,
  data = nimble_data,
```

```

  inits = initial_values,
  monitors = c("alpha", "sigma", "beta_exp", "gamma_size", "delta_loc", "sigma_loc"),
  niter = 50000, nburnin = 10000, nchains = 3,
  summary = TRUE, WAIC = TRUE
)

```

4.3 Summary table

The summary table below presents the results of our Bayesian model. It summarizes the posterior distributions for the key parameters.

For each parameter, we report the posterior mean, the posterior standard deviation, the 95% credible interval and the R-hat statistic, which is a diagnostic used to assess MCMC convergence; R-hat values very close to 1.0 indicate that the different chains have converged to the same posterior distribution, suggesting the results are stable and reliable.

```

summary_obj <- MCMCvis::MCMCsummary(mcmc_output$samples)

summary_clean <- summary_obj %>%
  tibble::rownames_to_column("Parameter") %>%
  filter(!grepl("^delta_loc", Parameter)) %>%
  select(Parameter, Mean = mean, StDev = sd, `2.5%`, `97.5%`, Rhat)

kbl(summary_clean, booktabs = TRUE,
     caption = "Posterior estimates") %>%
  kable_styling(latex_options = c("striped", "scale_down", "HOLD_position"))

```

Table 1: Posterior estimates

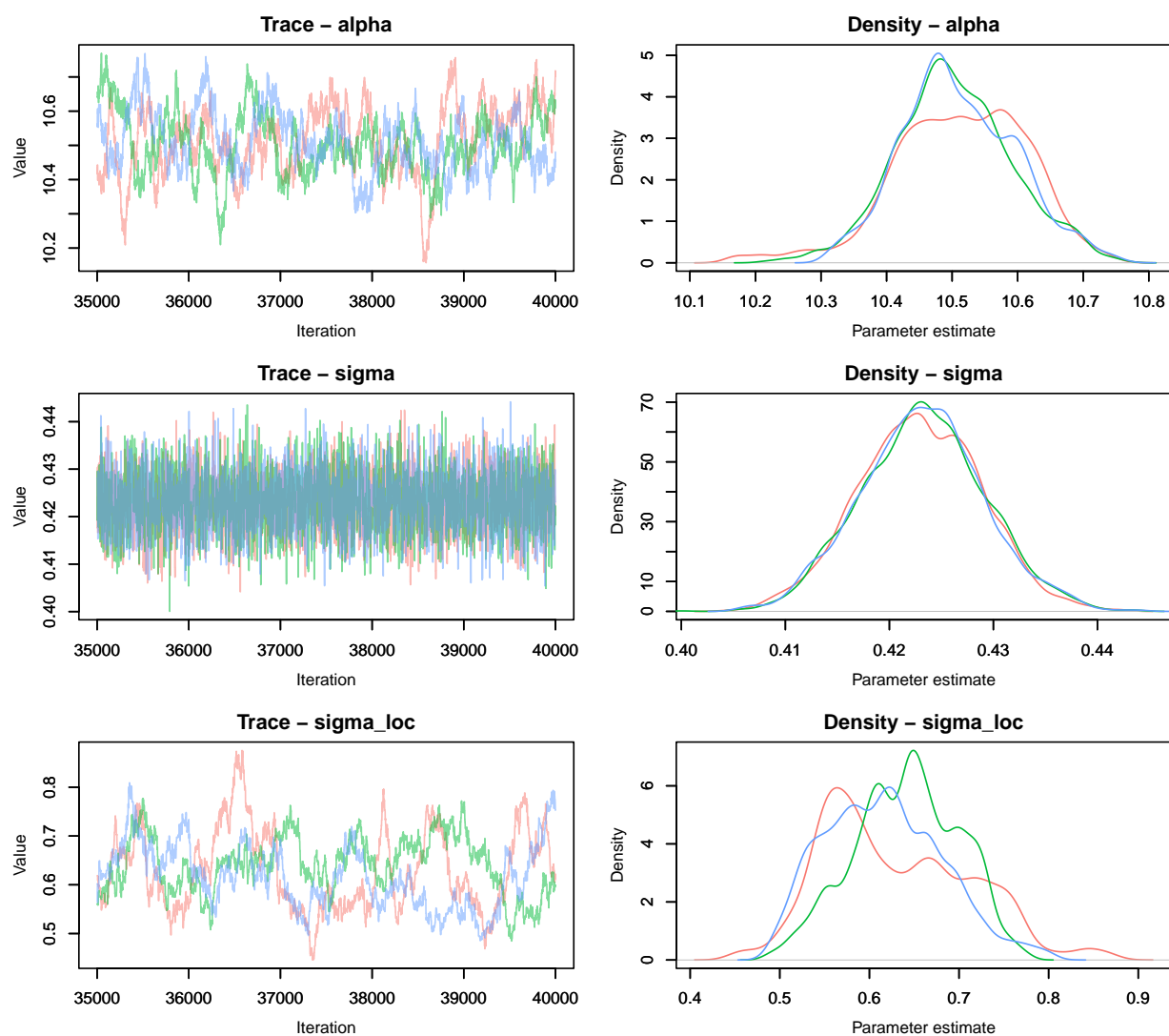
Parameter	Mean	StDev	2.5%	97.5%	Rhat
alpha	10.5263115	0.0884263	10.3556313	10.7046787	1.01
beta_exp[1]	0.0000000	0.0000000	0.0000000	0.0000000	NaN
beta_exp[2]	0.3385267	0.0323563	0.2750031	0.4021552	1.00
beta_exp[3]	0.6029923	0.0307135	0.5426114	0.6629646	1.00
beta_exp[4]	0.8461712	0.0523091	0.7432751	0.9478599	1.00
gamma_size[1]	0.0000000	0.0000000	0.0000000	0.0000000	NaN
gamma_size[2]	-0.0229014	0.0251965	-0.0722213	0.0270868	1.00
gamma_size[3]	-0.3056466	0.0439481	-0.3915918	-0.2195233	1.00
sigma	0.4229489	0.0059954	0.4113443	0.4348842	1.00
sigma_loc	0.6251588	0.0651535	0.5068692	0.7613072	1.02

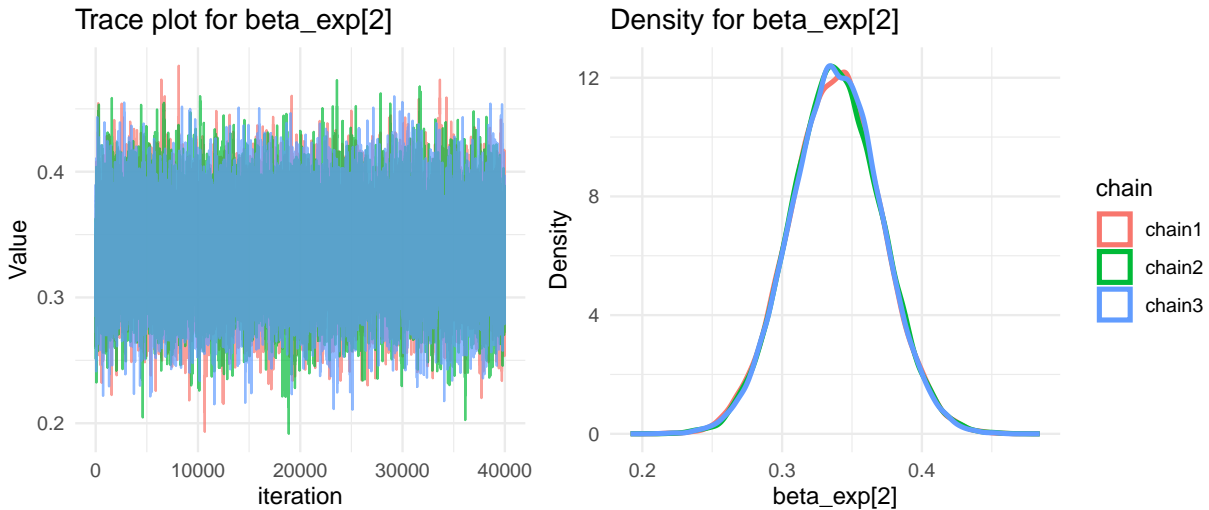
Our findings - Salaries rise strongly and credibly with experience level. - Medium companies pay roughly the same as small ones, but large companies surprisingly pay less. - Geographic location

introduces large variability in salaries, confirming the importance of modeling it as a random effect.
- The model converged well (R-hat values close to 1.0), so the estimates are trustworthy.

4.4 Trace plots

```
MCMCtrace(  
  object = mcmc_output$samples,  
  params = c("alpha", "sigma", "sigma_loc"),  
  ind = TRUE, pdf = FALSE  
)
```





The trace plots and posterior density plots provide a graphical visualisation of convergence and mixing across the three MCMC chains.

For the **intercept alpha**, the chains fluctuate around the same central value (approximately 10.5) with no visible trend.

For the **residual standard deviation sigma**, the traces are stable around 0.42, with excellent overlap between chains. For the **location-level variability sigma_loc**, the chains show slightly more variability and occasional differences, but they still explore the same posterior region (centered around 0.6). This is consistent with the slightly higher Rhat reported in the summary table (1.02), but overall convergence remains acceptable.

The trace plot for **beta_exp[2]** (mid-level vs. entry-level experience) shows stable fluctuations around 0.34.

The corresponding **density plots** confirm convergence; as it was the case before, `sigma_loc` shows broader densities but still with substantial overlap.

Overall, these plots indicate that the MCMC sampler has converged well, and the posterior estimates can be considered reliable for inference.

5. PPC

```
combined_draws <- do.call(rbind, mcmc_output$samples)
sel <- sample(1:nrow(combined_draws), size = 200)

yrep_mean <- numeric(length(sel))

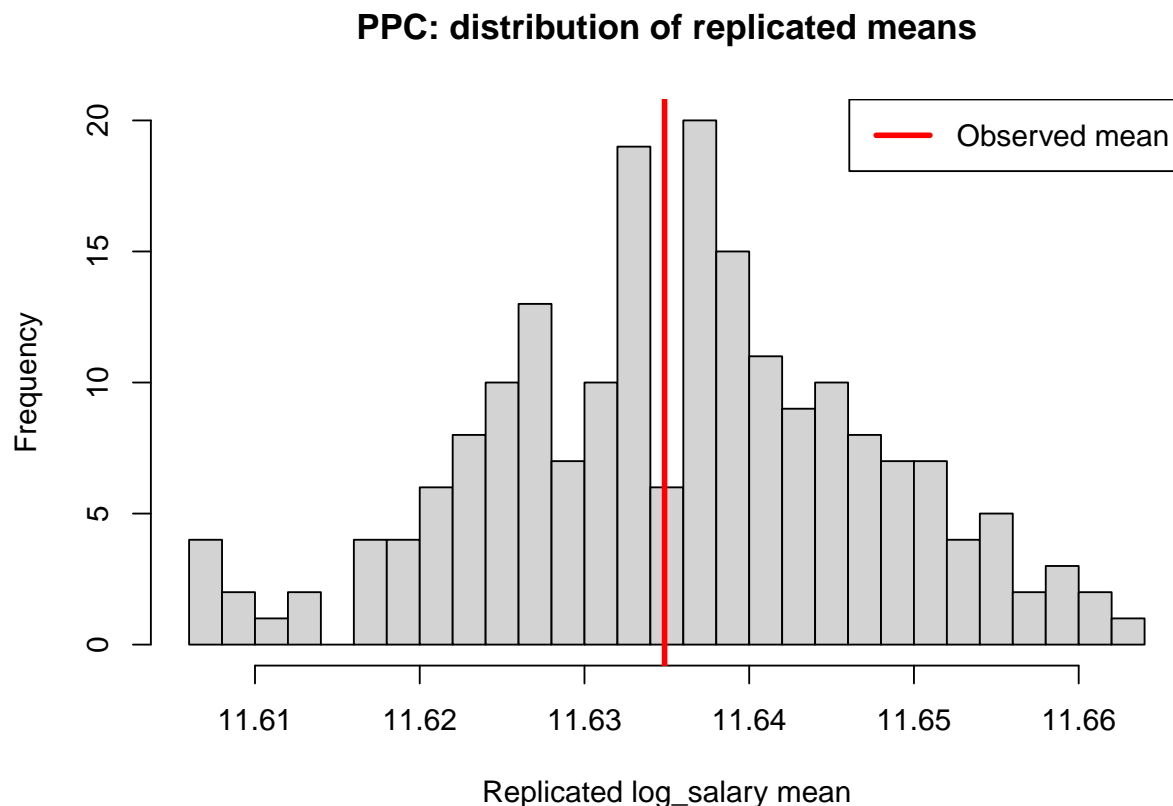
for (ii in seq_along(sel)) {
  draw <- combined_draws[sel[ii], ]
  mu_vec <- draw["alpha"] +
    draw[grep("^beta_exp\\[", names(draw))][experience_level_int] +
```

```

draw[grep("^gamma_size\\\[", names(draw))][company_size_int] +
draw[grep("^delta_loc\\\[", names(draw))][company_location_int]
y_rep <- rnorm(N, mean = mu_vec, sd = draw["sigma"])
yrep_mean[ii] <- mean(y_rep)
}

hist(yrep_mean, main = "PPC: distribution of replicated means",
      xlab = "Replicated log_salary mean", breaks = 30)
abline(v = mean(y), col = "red", lwd = 3)
legend("topright", "Observed mean", col = "red", lty = 1, lwd = 3)

```



To evaluate the adequacy of the model, we performed a posterior predictive check (PPC): New datasets were simulated from the posterior distribution by drawing parameter values from the MCMC output and generating replicated salaries. For each simulated dataset, we computed the mean of the replicated log-salaries.

The histogram shows the distribution of these replicated means across simulations, while the vertical red line marks the observed mean log-salary in the actual data.

The observed mean falls very close to the center of the replicated distribution, indicating that the model is able to reproduce the main feature of the data (the average log-salary). This suggests that the model provides a good fit at the level of the mean structure, supporting the validity of the hierarchical specification.

6. Conclusion

This analysis explored the factors that influence salaries in data science using a Bayesian model. The results confirm that **experience level** is the main driver of differences in pay, with clear increases from entry-level to executive positions. **Company size** also matters, although the effect is less straightforward, with only large companies showing a noticeable difference.

By treating **company location** as a random effect, the model was able to capture substantial variation across countries. This highlights how strongly geography shapes salaries, even after accounting for individual characteristics like experience and company size.

The model diagnostics support the reliability of these findings. The trace plots and R-hat statistics show good convergence of the MCMC chains, while the posterior predictive check suggests that the model is able to reproduce the main structure of the observed data. Overall, the Bayesian approach offered a flexible and transparent way to study salary patterns, providing both estimates of the effects and a clear quantification of uncertainty. These results underline the importance of experience and geography in shaping compensation in the global data science job market.