Midterm: Examing Gender Pay Gap with Bayesian Regression

Emma Kruis

2020-02-01

Instructions

This script reviews *Bayesian Regression Analysis* as part of the *Midterm Review*. You will use content from the lecture and assignment materials on *Bayesian Regression Analysis* to complete this script. You will *copy and paste* relevant code from those files into this script and answer the associated questions for each task. You will respond to questions in each section after executing relevant code to answer a question. You will submit this script to its *Submissions* folder on *D2L* as part of the *Midterm Review*. For this script, you will submit *two* files:

1. this completed *R Markdown* script, and
2. as a first preference, a *PDF* (if you already installed `TinyTeX` properly), as a second preference, a *Microsfot Word* (if your computer has *Microsoft Word*) document, or, as a third preference, an *HTML* (if you did *not* install `TinyTeX` properly and your computer does *not* have *Microsoft Word*) file to *D2L*.

For the *Midterm Review*, create the project directory: *~/mgt_592/assignments/midterm_review*. Convert your project directory into a formal *R Project* directory by going to the *File* menu in *RStudio*, selecting *New Project...*, choosing *Existing Directory*, and going to your *~/mgt_592/assignments/midterm_review* folder to select it as the top-level directory for this **R Project**.
The project directory should contain the following folders: *scripts*, *data*, and *plots*. Store this script in the *scripts* folder and the relevant data in the *data* folder.

Global Settings

The first code chunk sets the global settings for the remaining code chunks in the document. Do *not* change anything in this code chunk.

Task 1: Load Libraries

For this task, you will load the libraries you need for this script.

Task 1.1

In this code chunk, load the following packages:

1. **here**,
2. **tidyverse**,
3. **janitor**,
4. **skimr**,

5. **ggthemes**,
6. **rstanarm**,
7. **bayesplot**, and
8. **tidybayes**.

Make sure you installed these packages before loading the libraries.

You will use functions from these packages to complete this script.

```
### load libraries for use in current working session
## here for project work flow
library(here)
```

```
## here() starts at /Users/emmakruis/Library/Mobile Documents/com~apple~CloudDocs/year_2/WQ21/mgt_592/as
```

```
## tidyverse for data manipulation and plotting
## loads eight different libraries simultaneously
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## janitor to clean data and chi-square test
library(janitor)
```

```
##
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
## skimr to summarize data
library(skimr)

## ggthemes for plot themes
library(ggthemes)

## rstanarm for Bayesian regression
library(rstanarm)
```

```
## Loading required package: Rcpp
```

```
## This is rstanarm version 2.21.1

## - See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

## - Default priors may change, so it's safest to specify priors, even if equivalent to the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

##   options(mc.cores = parallel::detectCores())
```

```
## bayesplot for Bayesian posterior distributions
library(bayesplot)
```

```
## This is bayesplot version 1.8.0

## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting
```

```
## tidybayes to work with Bayesian model results
library(tidybayes)
```

Task 2: Import Data

For this task, you will import the data file: **gender_pay_gap.csv**.

Task 2.1

Use the **read_csv()** and **here()** functions to load the data file for this working session. Save the data as the object **org_raw**.

Make a copy of the data and name the copy: **org_work**. You will work with the *complete data*. Use the **glimpse()** function to view a preview of values for each variable in **org_work**.

```
org_raw<- read_csv(
  here("data", "gender_pay_gap.csv")
)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   jobTitle = col_character(),
##   gender = col_character(),
##   age = col_double(),
##   perfEval = col_double(),
```

```
##   edu = col_character(),
##   dept = col_character(),
##   seniority = col_double(),
##   basePay = col_double(),
##   bonus = col_double()
## )
```

```
org_work <- org_raw


glimpse(org_work)
```

```
## Rows: 1,000
## Columns: 9
## $ jobTitle  <chr> "Graphic Designer", "Software Engineer", "Warehouse Associat~
## $ gender    <chr> "Female", "Male", "Female", "Male", "Male", "Female", "Femal~
## $ age       <dbl> 18, 21, 19, 20, 26, 20, 20, 18, 33, 35, 24, 18, 19, 30, 35, ~
## $ perfEval  <dbl> 5, 5, 4, 5, 5, 5, 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ edu       <chr> "College", "College", "PhD", "Masters", "Masters", "PhD", "C~
## $ dept      <chr> "Operations", "Management", "Administration", "Sales", "Engi~
## $ seniority <dbl> 2, 5, 5, 4, 5, 4, 4, 5, 5, 5, 5, 3, 3, 5, 4, 3, 5, 5, 5, 5, ~
## $ basePay   <dbl> 42363, 108476, 90208, 108080, 99464, 70890, 67585, 97523, 11~
## $ bonus     <dbl> 9938, 11128, 9268, 10154, 9319, 10126, 10541, 10240, 9836, 9~
```

Task 3: Clean and Prepare Data

For this task, you will clean and prepare the data.

Task 3.1

Perform the following cleaning tasks to update **org_work**:

1. mutate all character variables to factor variables, and
2. use **clean_names()** to make variable names use all lowercase letters and connect multiple words with underscores.

Use **glimpse()** to preview the updated **org_work** data object.

```
org_work <- org_work %>%
  mutate(
    across(
      .cols = where(is.character),
      .fns = as_factor
    )
  ) %>%
  clean_names()

glimpse(org_work)
```

```
## Rows: 1,000
## Columns: 9
```

```
## $ job_title <fct> Graphic Designer, Software Engineer, Warehouse Associate, So~
## $ gender    <fct> Female, Male, Female, Male, Male, Female, Female, Male, Fema~
## $ age       <dbl> 18, 21, 19, 20, 26, 20, 20, 18, 33, 35, 24, 18, 19, 30, 35, ~
## $ perf_eval <dbl> 5, 5, 4, 5, 5, 5, 5, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ edu       <fct> College, College, PhD, Masters, Masters, PhD, College, PhD, ~
## $ dept      <fct> Operations, Management, Administration, Sales, Engineering, ~
## $ seniority <dbl> 2, 5, 5, 4, 5, 4, 4, 5, 5, 5, 5, 3, 3, 5, 4, 3, 5, 5, 5, 5, ~
## $ base_pay  <dbl> 42363, 108476, 90208, 108080, 99464, 70890, 67585, 97523, 11~
## $ bonus     <dbl> 9938, 11128, 9268, 10154, 9319, 10126, 10541, 10240, 9836, 9~
```

Task 4: Examine Data

For this task, you will examine the data.

Task 4.1

Create a plot using **ggplot()** to highlight the differences in *base pay* between *men* and *women* for different *departments*.

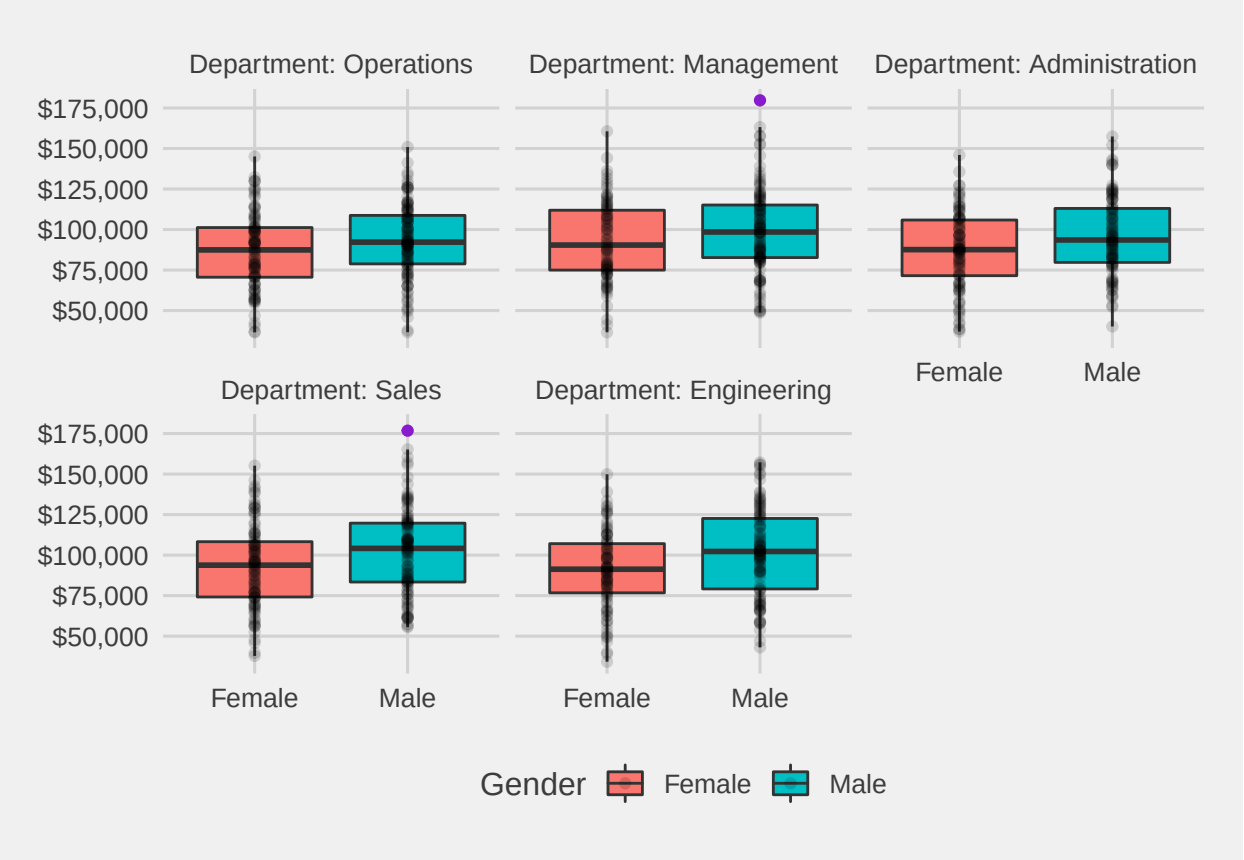
Do the following:

1. call **ggplot()**, set the data to **org_work**, map **gender** to the *x-axis* and the **fill**, and map **base_pay** to the *y-axis*;
2. call **geom_boxplot()** and set the outlier **color** to **purple** and **size** to **1.5**;
3. call **geom_point()** and set **alpha** to **0.15**;
4. call **facet_wrap()** to facet by **dept** with **2** rows;
5. call **scale_y_continuous** and set the number of breaks to **8** and the **labels** to *dollar format*;
6. label the axes and legend appropriately with **labs()**;
7. use **theme_fivethirtyeight()** as the *theme*.

Questions 4.1: Answer these questions: (1) For which combinations of *department* and *gender* are there outliers? (2) Is the *median total pay* higher for *women* in any of the *department* groups?

Responses 4.1: *(1) management and sales (2) No* .

```
ggplot(
  org_work,
  aes(x =gender, y = base_pay, fill = gender)
) +
  geom_boxplot(outlier.color = "purple", outlier.size = 1.5) +
  geom_point(alpha = 0.15) +
  facet_wrap(
    vars(dept),
    nrow = 2,
    labeller = as_labeller(
      setNames(
      paste("Department", levels(org_work$dept), sep = ": "),
      levels(org_work$dept)
      )
    )
  )+
  scale_y_continuous(n.breaks = 8, labels = scales::dollar_format()) +
  labs(x = "Gender", Y = "Base Pay", fill= "Gender") +
  theme_fivethirtyeight()
```



Task 5: Fit Moderated Bayesian Regression Model

Estimate a simple Bayesian regression model using using **org_work** where observed values of *base pay* are predicted from observed values of *gender* and *department*.

Task 5.1

Create a model object named **mod_1** using **stan_glm()**. Inside of **stan_glm()**, do the following:

1. specify the *formula* to indicate **base_pay** predicted by **gender**, **dept**, and their interaction,
2. specify the *data* as **org_work**,
3. set the *prior intercept* to a **normal** distribution with **location** equal to **8e4** and **autoscale** equal to **TRUE**,
4. set the *prior of the regression coefficients for the predictors* to a **normal** distribution with **location** equal to **0** and **autoscale** equal to **TRUE**,
5. set the *error prior* to be an **exponential** distribution with **rate** equal to **0.7** and **autoscale** equal to **TRUE**, and
6. set the **seed** to **1805** (birth year of *William Rowan Hamilton*).

After creating the model, then do the following:

1. apply **summary()** to **mod_1** using *three digits* and **seq(0.1, 0.9, 0.2)** for the *percentiles* to examine the posterior parameter distributions,
2. apply **coef()** to **mod_1** to extract the *median* regression coefficients, and

3. apply **posterior_interval** to **mod_1** and set the *credible interval* to **0.89**.

Questions 5.1: Answer these questions: (1) What is the *90th percentile* of the **genderMale:deptAdministration** regression coefficient? (2) What is the estimated *median difference in base pay* between *men* and *women* in the *operations* department (i.e., look at the **genderMale** regression coefficient)? (3) What is the *89% credible interval* for the difference in pay for *women* in the *management department* versus the *operations department* (i.e., look at the **deptManagement** row)? (4) What is the *89% credible interval* for the difference in pay for *men* versus *women* in the *management department* (i.e., look at the **genderMale:deptManagement** row)?

Responses 5.1: *(1) 10618.898 (2) 8537.523 (3) 91825.96 (4) 11116.88.*

```
##model creation
mod_1 <- stan_glm(
  base_pay ~ gender,
  data = org_work,
  prior_intercept = normal(location= 8e+04, autoscale = TRUE),
  prior = normal(location = 0, autoscale = TRUE),
  prior_aux = exponential(rate = 0.7, autoscale = TRUE),
  seed = 1805
)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 7.5e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.75 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 1.11387 seconds (Warm-up)
## Chain 1:                0.102827 seconds (Sampling)
## Chain 1:                1.21669 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 1.8e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.18 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
```

```
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 1.07515 seconds (Warm-up)
## Chain 2:                0.106884 seconds (Sampling)
## Chain 2:                1.18203 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1.7e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.17 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 1.13779 seconds (Warm-up)
## Chain 3:                0.100497 seconds (Sampling)
## Chain 3:                1.23828 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 1.3e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.13 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 1.62348 seconds (Warm-up)
## Chain 4:                0.100787 seconds (Sampling)
## Chain 4:                1.72427 seconds (Total)
## Chain 4:
```

```
##model summary
summary(mod_1, digits = 3, probs = seq(0.1, 0.9,0.2))
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      base_pay ~ gender
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 1000
##  predictors:   2
##
## Estimates:
##               mean      sd        10%       30%       50%       70%
## (Intercept) 89934.675  1189.709 88417.816 89330.944 89916.702 90559.858
## genderMale   8520.215  1638.351  6391.848  7665.927  8537.523  9384.258
## sigma       25003.548   552.191 24297.530 24707.403 24994.065 25273.245
##               90%
## (Intercept) 91451.990
## genderMale  10618.898
## sigma       25727.433
##
## Fit Diagnostics:
##            mean      sd        10%       30%       50%       70%       90%
## mean_PPD 94458.737  1110.277 93044.828 93871.391 94436.555 95028.815 95889.679
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse   Rhat   n_eff
## (Intercept)   19.575  1.000 3694
## genderMale    27.458  1.000 3560
## sigma          8.547  1.000 4174
## mean_PPD      17.320  1.000 4109
## log-posterior  0.029  1.001 1847
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

```
##extracting coefficients
coef(mod_1)
```

```
## (Intercept)  genderMale
##   89916.702    8537.523
```

```
##examine credible intervals
posterior_interval(mod_1, prob = 0.89)
```

```
##                   5.5%    94.5%
## (Intercept) 88050.621 91825.96
## genderMale   5871.647 11116.88
## sigma       24134.374 25910.79
```

Task 5.2

Examine **mod_1** by doing the following:

1. compute the *Bayesian R-squared* using **bayes_R2()** and saving the results to an object named **mod_1_r_sq**;
2. apply **summary()** to **mod_1_r_sq**;
3. apply **pp_check()** and examine the *density overlay*, the *mean* on its own, and the *mean* and *sd* together;
4. compute the *fitted values* using the *median posterior regression parameters* and save the calculation as a new variable named **mod_1_fitted** to **org_work**;
5. open the *spreadsheet view* of **org_work** to answer a question about **mod_1_fitted** values.

Questions 5.2: Answer these questions: (1) What is the *median R-squared* value? (2) Do the *posterior predictive checks* indicate any estimation issues? (3) Use the *spreadsheet view* of **org_work** to answer: what is the *median posterior fitted value* for a *female* employee working in the *engineering* department? (4) Use the *spreadsheet view* of **org_work** to answer: what is the *median posterior fitted value* for a *male* employee working in the *engineering* department?

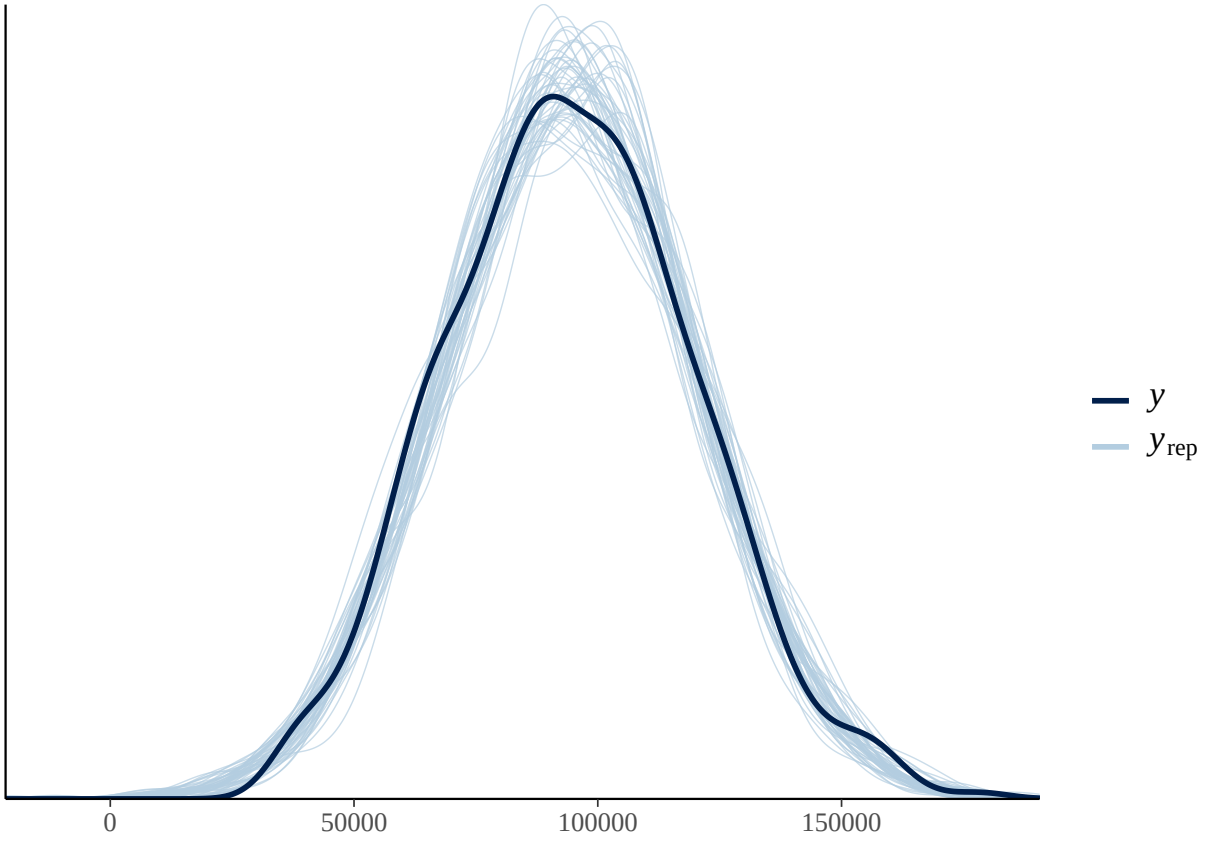
Responses 5.2: *(1) 0.028287 (2) No estimation issues (3) 89916.70 (4) 98454.23.*

```
mod_1_r_sq <- bayes_R2(mod_1)

summary(mod_1_r_sq)
```

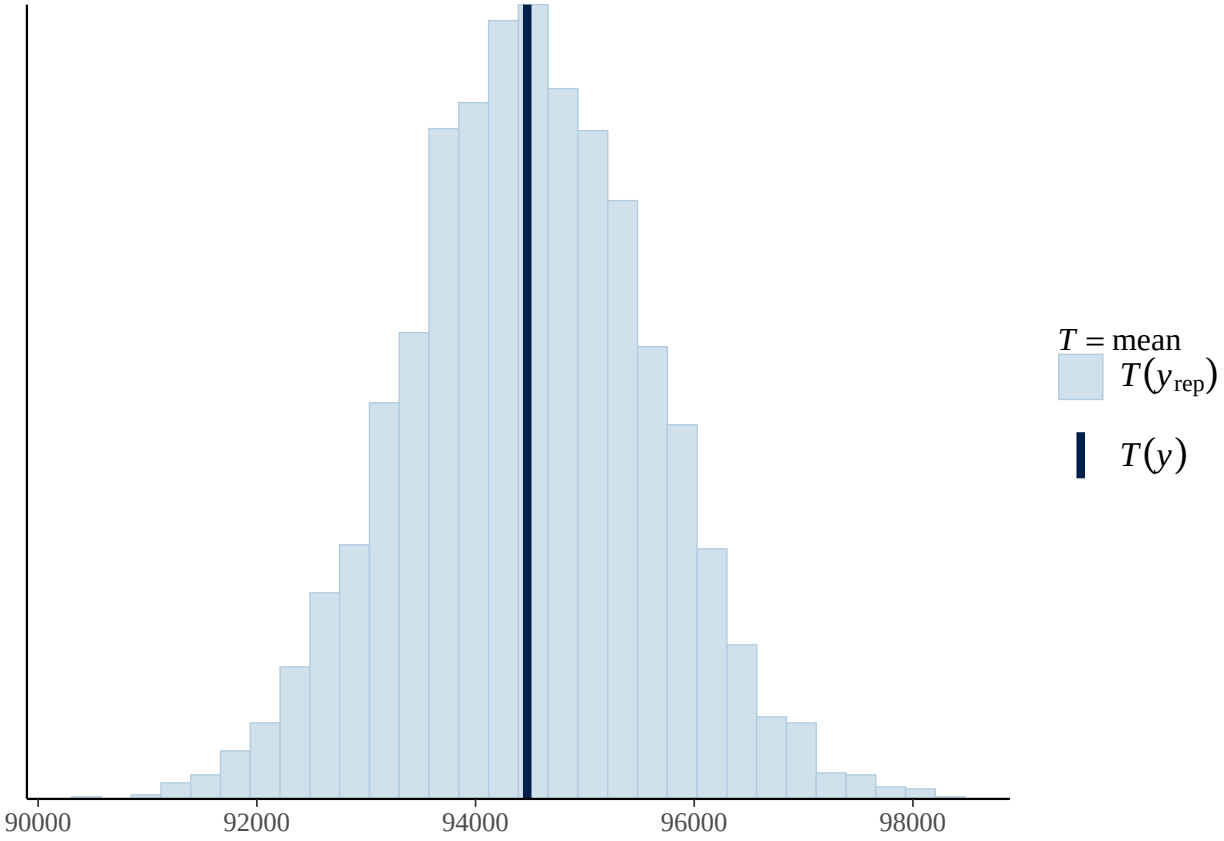
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.002516 0.021392 0.028287 0.029055 0.035814 0.085586
```

```
##posterior predictive checks
#density overlay
pp_check(mod_1, "dens_overlay")
```

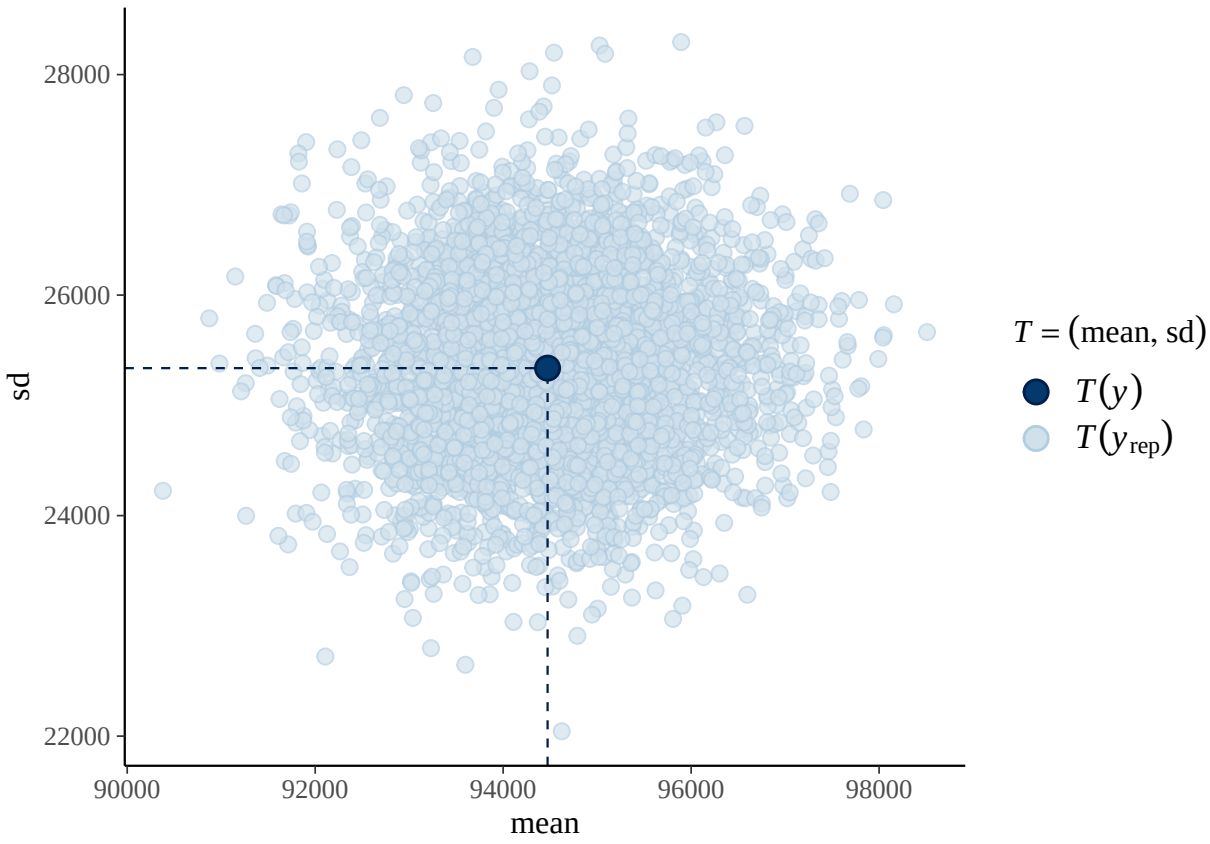



```
#mean
pp_check(mod_1, plotfun = "stat", stat= "mean")
```

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




```
#mean and sd
pp_check(mod_1, plotfun = "stat_2d", stat = c("mean", "sd"))
```




```
##median posterior fitted values
org_work <- org_work %>%
  mutate(mod_1_fitted = fitted(mod_1))

mod_1_fitted <- as_tibble(mod_1)


mod_1_fitted
```

```
## # A tibble: 4,000 x 3
##    `(Intercept)` genderMale  sigma
##            <dbl>      <dbl>  <dbl>
##  1        89798.     10372. 25106.
##  2        89922.      6918. 24851.
##  3        89598.     10327. 24735.
##  4        90122.      6730. 25221.
##  5        90306.      6481. 25048.
##  6        91181.      7077. 25449.
##  7        89249.     10906. 25159.
##  8        90700.      7401. 24459.
##  9        90984.      7101. 24801.
## 10        89417.      9954. 24506.
## # ... with 3,990 more rows
```

Task 5.3

Extract the draws from the posterior regression parameter distributions from **mod_1** with **as_tibble()** and save them as **mod_1post**. Create a plot using **mcmc_areas()**. Do the following to create the plot:

1. call **mcmc_areas()** and set data to **mod_1_post**, select the **`genderMale:deptSales`** (note the back ticks) parameter, and set the interval to **0.89**;
2. call **ggtitle()** and set the *title* of the plot to **Posterior Distribution for Gender Difference in Base Pay in Sales Department**.

Question 5.3: What can you say about the *gender difference in base pay in the sales department*?

Response 5.3: *This confirms that there is a gender pay gap even at the 100th percentile.* .

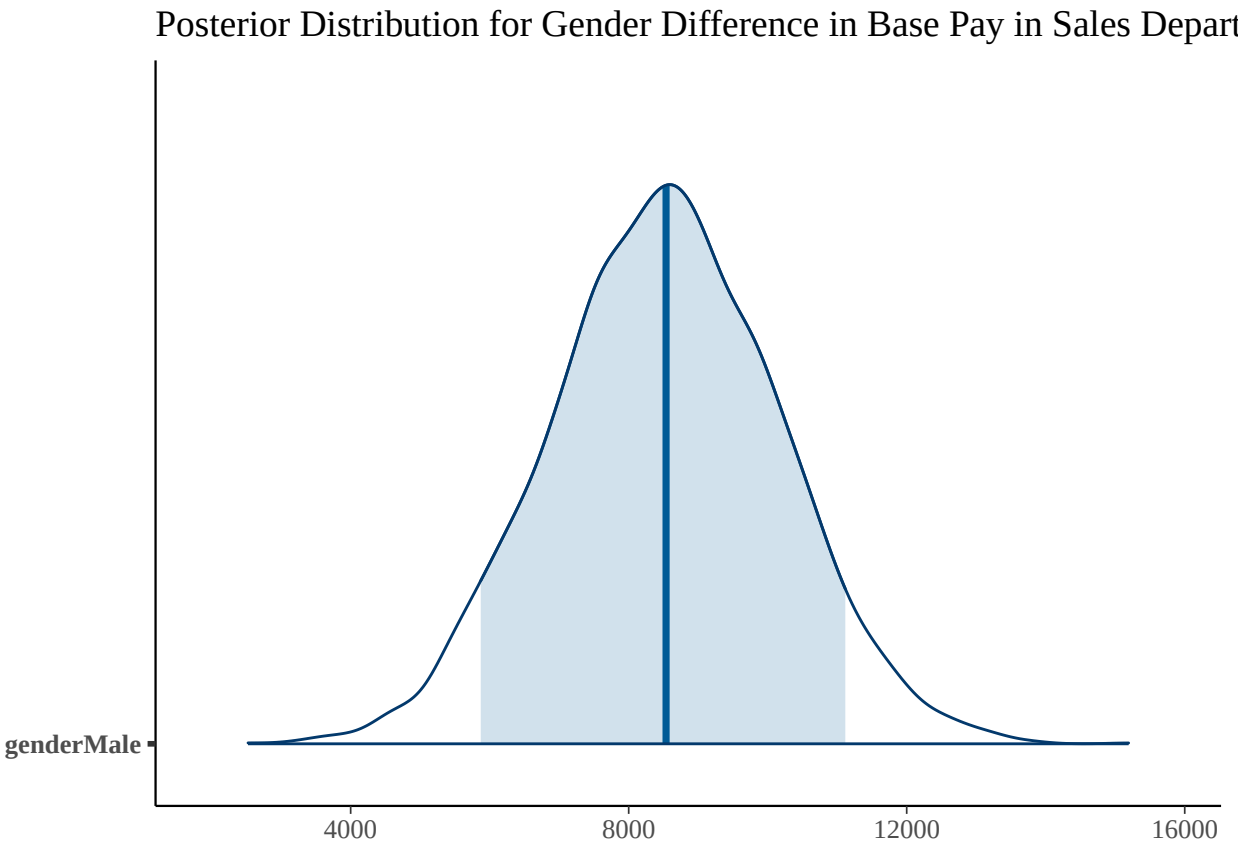
```
mod_1_post <- as_tibble(mod_1)

mod_1_post
```

```
## # A tibble: 4,000 x 3
##    `(Intercept)` genderMale  sigma
##            <dbl>      <dbl>  <dbl>
##  1        89798.     10372. 25106.
##  2        89922.      6918. 24851.
##  3        89598.     10327. 24735.
##  4        90122.      6730. 25221.
##  5        90306.      6481. 25048.
##  6        91181.      7077. 25449.
##  7        89249.     10906. 25159.
##  8        90700.      7401. 24459.
##  9        90984.      7101. 24801.
## 10        89417.      9954. 24506.
## # ... with 3,990 more rows
```

```
#posterior parmeter distributions
mcmc_areas(
  mod_1_post,
  pars = vars(genderMale),
  prob = 0.89
)+

  ggtitle("Posterior Distribution for Gender Difference in Base Pay in Sales Department")
```



Task 5.4

Calculate the *fitted* and *predicted* values from the *posterior draws* of the regression parameters where you set **gender** equal to **levels(org_work***gender*)∗∗(*i.e.*, ∗∗*gender* = *levels*(*org*$_w$*ork***gender)**) and **dept** equal to **levels(org_work***dept*)∗∗(*i.e.*, ∗∗*dept* = *levels*(*org*$_w$*ork***gender)**) inside of **crossing()**. Pass the **crossing()** result to **add_fitted_draws()** and **add_predicted_draws()** while including **mod_1** as an input as well for the *fitted* and *predicted* values saving the results as **mod_1_post_fit** and **mod_1_post_pred**, respectively.

Create a plot named **mod_1_plot** using **ggplot()**. To create the plot, do the following:

1. inside of **ggplot()**, set data to **org_work**, **gender** to the *x-axis*, and **base_pay** to the *y-axis*;
2. add a first **geom_jitter()** layer with **height** and **width** set to **0.05** and **alpha** set to **0.5**.
3. add a second **geom_jitter()** layer with **height** and **width** set to **0.05**, set data to **mod_1_post_pred**, map **gender** to the *x-axis* and **.prediction** to the *y-axis*, set **color** to **lightgreen**, set **size** to **0.5**, and set **alpha** to **0.15**;
4. add a third **geom_jitter()** layer with **height** and **width** set to **0.05**, set data to **mod_1_post_fit**, map **gender** to the *x-axis* and **.value** to the *y-axis*, set **color** to **skyblue**, set **size** to **1.5**, and set **alpha** to **0.15**;
5. add a **geom_point()** layer and map **mod_1_fitted** to the *y-axis*, set **color** to **red**, and set **size** to **2**;
6. add **facet_wrap()** to facet by **dept** across **2** rows;
7. call **scale_y_continuous** and set the number of breaks to **10** and the **labels** to *dollar format*;
8. label the x-axis **Gender** and the y-axis **Base Pay**;
9. use **theme_hc()**.

Display **mod_1_plot** by typing its name.

Questions 5.4: Answer these questions: (1) Do *men* outearn *women* irrespective of *department*? (2) Is there much of an *interaction effect* between *gender* and *department*? (Hint: you can also examine the *89% credible intervals* for the interaction regression coefficients to answer this question.)

Responses 5.4: *(1) Yes (2) Yes* .

```
mod_1_post_fit <- crossing(
  gender = levels(org_work$gender),
  dept = levels(org_work$gender)
  ) %>%
  add_fitted_draws(mod_1)
```

```
## Instead of posterior_linpred(..., transform=TRUE) please call posterior_epred(), which provides equi
```

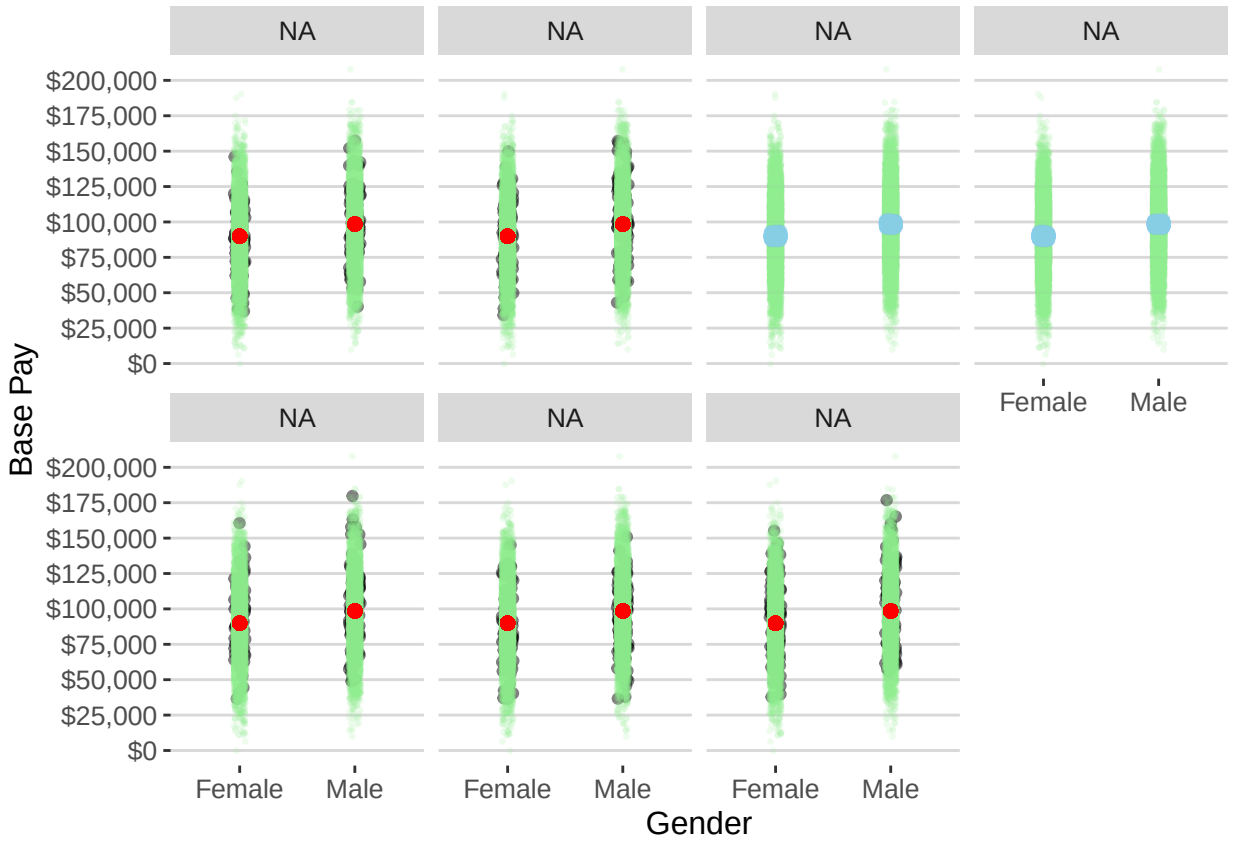
```
mod_1_post_pred <- crossing(
  gender = levels(org_work$gender),
   age = levels(org_work$gender)
  ) %>%
  add_predicted_draws(mod_1)



mod_1_plot <- ggplot(
  # data
  org_work,
  # mapping
  aes(x = gender, y = base_pay)
  ) +
  ## jitter observed data points
  geom_jitter(height = 0.05, width = 0.05, alpha = 0.5) +
  ## add posterior predicted values
  geom_jitter(
    # adjust height, width
    height = 0.05, width = 0.05,
    # data
    data = mod_1_post_pred,
    # mapping
    mapping = aes(x = gender, y = .prediction),
    # color, size, alpha
    color = "lightgreen", size = 0.5, alpha = 0.15
  ) +
  ## add posterior fitted values
  geom_jitter(
    # adjust height, width
    height = 0.05, width = 0.05,
    # data
    data = mod_1_post_fit,
    # mapping
    mapping = aes(x = gender, y = .value),
    # color, size, alpha
    color = "skyblue", size = 1.5, alpha = 0.15
  ) +
  ## add median posterior fitted values
  geom_point(
```

```
    # map fitted values
    aes(y = mod_1_fitted),
    # color, size
    color = "red", size = 2
  ) +
   facet_wrap(
    # facet by variable,
    vars(dept),
    # display across rows
    nrow = 2,
    # labels
    labeller = as_labeller(
      # look-up table
      setNames(
        # vector elements
        paste("Department", 1:5, sep = ": "),
        # names of elements
        1:5
      )
    )
  ) +
  ## scale y-axis
  scale_y_continuous(n.breaks = 10, labels = scales::dollar_format()) +
  ## labels
  labs(x = "Gender", y = "Base Pay") +
  #define theme
  theme_hc()

## display plot
mod_1_plot
```



Task 6: Save Object

For this task, you will save a plot.

Task 6.1

Save **mod_1_plot** as **bayes_reg.png** in the **plots** folder of the project directory using **ggsave()**. Use a width of *9 inches* and height of *6 inches* for all plots.

```
#mod_1_plot
ggsave(
  here("plots", "bayes_reg.png"),
  plot = mod_1_plot,
  units = "in", width = 9, height = 6
)
```

Task 7: Conceptual Question

For your last task, you will respond to a conceptual question.

Question 7.1: What is the difference between the *fitted* and *predicted* values produced by **add_fitted_draws()** and **add_predicted_draws()**, respectively?

Response 7.1: *Fitted values adds draws from posterior liner PREDICTORS to the data. Predicted values adds draws from posterior PREDICTIONS to the data. . .*