tutorial7-completed.R

garethmoen

2021-11-19

```r
##############################################
# Tutorial 7: Multiple Linear Regression in R
##############################################

#### Goals:
#### 1. Learn the different methods for MLR in R
#### 2. Learn how to organise regression models
#### 3. Create workflows through to visualisation

options(scipen = 999)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
#install.packages("broom")
library(broom)
?broom

#################
# Loading in data
#################

# read in the following url as "salary":
# https://raw.githubusercontent.com/ASDS-TCD/StatsI_Fall2021/main/datasets/salary.csv

salary <- read_csv("https://raw.githubusercontent.com/ASDS-TCD/StatsI_Fall2021/main/datasets/salary.csv
```

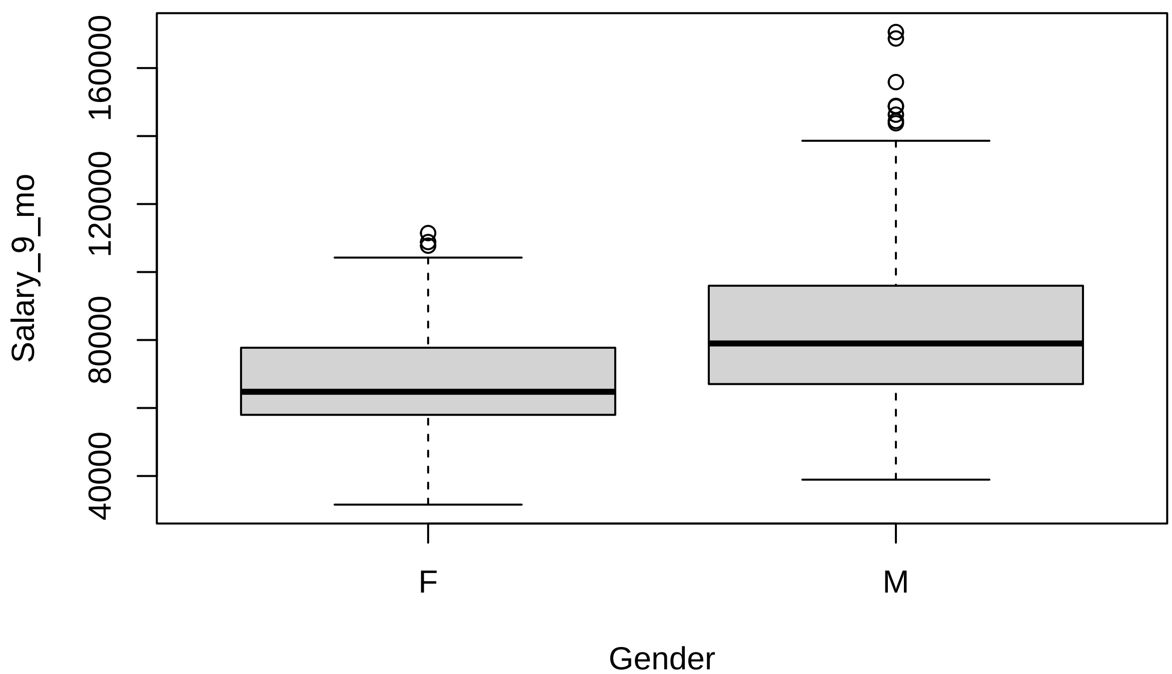
```
## New names:
## * `` -> ...1
## Rows: 424 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Department, Gender
## dbl (5): ...1, X, Rank_Code, Salary_9_mo, Avg_Cont_Grants
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#####
# EDA
#####

# Run a quick exploratory data analysis of the salary dataset. How does salary vary
# according to gender? How would we quicky visualise this? What about a test of
# significance?

# Summary stats
summary(salary)
```

```
##       ...1              X          Department           Rank_Code    
##  Min.   :  1.0   Min.   :  1.0   Length:424         Min.   :1.000  
##  1st Qu.:107.8   1st Qu.:146.8   Class :character   1st Qu.:1.000  
##  Median :213.5   Median :295.5   Mode  :character   Median :1.000  
##  Mean   :214.0   Mean   :298.4                      Mean   :1.649  
##  3rd Qu.:320.2   3rd Qu.:445.2                      3rd Qu.:2.000  
##  Max.   :427.0   Max.   :602.0                      Max.   :3.000  
##     Gender           Salary_9_mo     Avg_Cont_Grants  
##  Length:424         Min.   : 31582   Min.   :    600  
##  Class :character   1st Qu.: 64942   1st Qu.:  54804  
##  Mode  :character   Median : 76851   Median : 159346  
##                     Mean   : 81606   Mean   : 336714  
##                     3rd Qu.: 94370   3rd Qu.: 403425  
##                     Max.   :170591   Max.   :2330706
```

```r
# Base boxplot
boxplot(Salary_9_mo ~ Gender, data = salary)
```



```r
# Find means using [] subsetting
mean(salary$Salary_9_mo[salary$Gender == "M"])
```

```
## [1] 83898.26
```

```r
mean(salary$Salary_9_mo[salary$Gender == "F"])
```

```
## [1] 69602.34
```

```r
# Find means using pipe and dplyr
salary %>%
  group_by(Gender) %>%
  summarise(mean = mean(Salary_9_mo))
```

```
## # A tibble: 2 x 2
##   Gender   mean
##   <chr>   <dbl>
## 1 F      69602.
## 2 M      83898.
```

```r
# How about a quick check for statistical significance?
t.test(salary$Salary_9_mo ~ salary$Gender, mu = 0)
```

```
##
##  Welch Two Sample t-test
##
## data:  salary$Salary_9_mo by salary$Gender
## t = -6.0595, df = 118.56, p-value = 0.00000001661
## alternative hypothesis: true difference in means between group F and group M is not equal to 0
## 95 percent confidence interval:
##  -18967.628  -9624.207
## sample estimates:
## mean in group F mean in group M
##        69602.34        83898.26
```

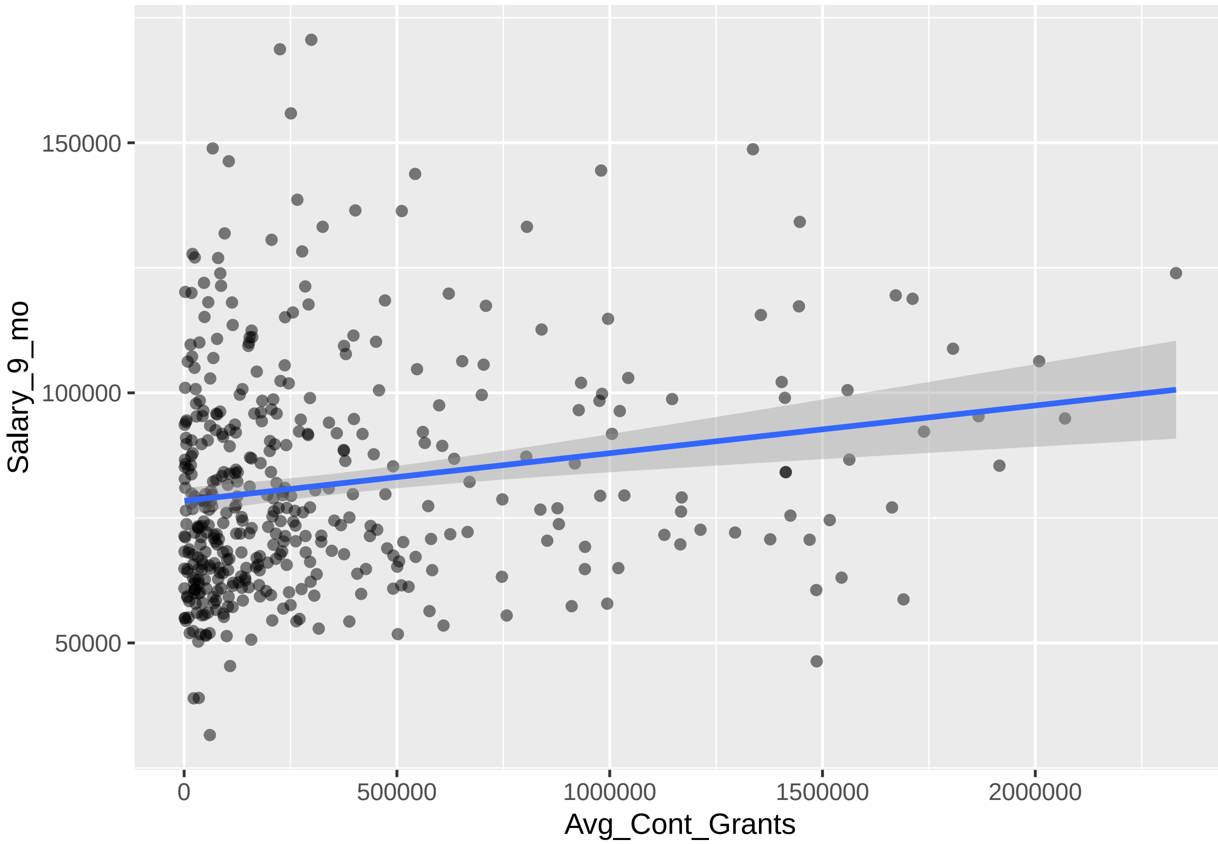
```r
# How do we interpret our test?


######################
# Running a regression
######################

# Is winning grants associated with salary? How would we find out? Can we visualise
# this relationship?

lm(Salary_9_mo ~ Avg_Cont_Grants, data = salary)
```

```
##
## Call:
## lm(formula = Salary_9_mo ~ Avg_Cont_Grants, data = salary)
##
## Coefficients:
##     (Intercept)  Avg_Cont_Grants
##    78394.083392         0.009538
```

```r
ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo)) +
  geom_point(alpha = 0.5) +
```

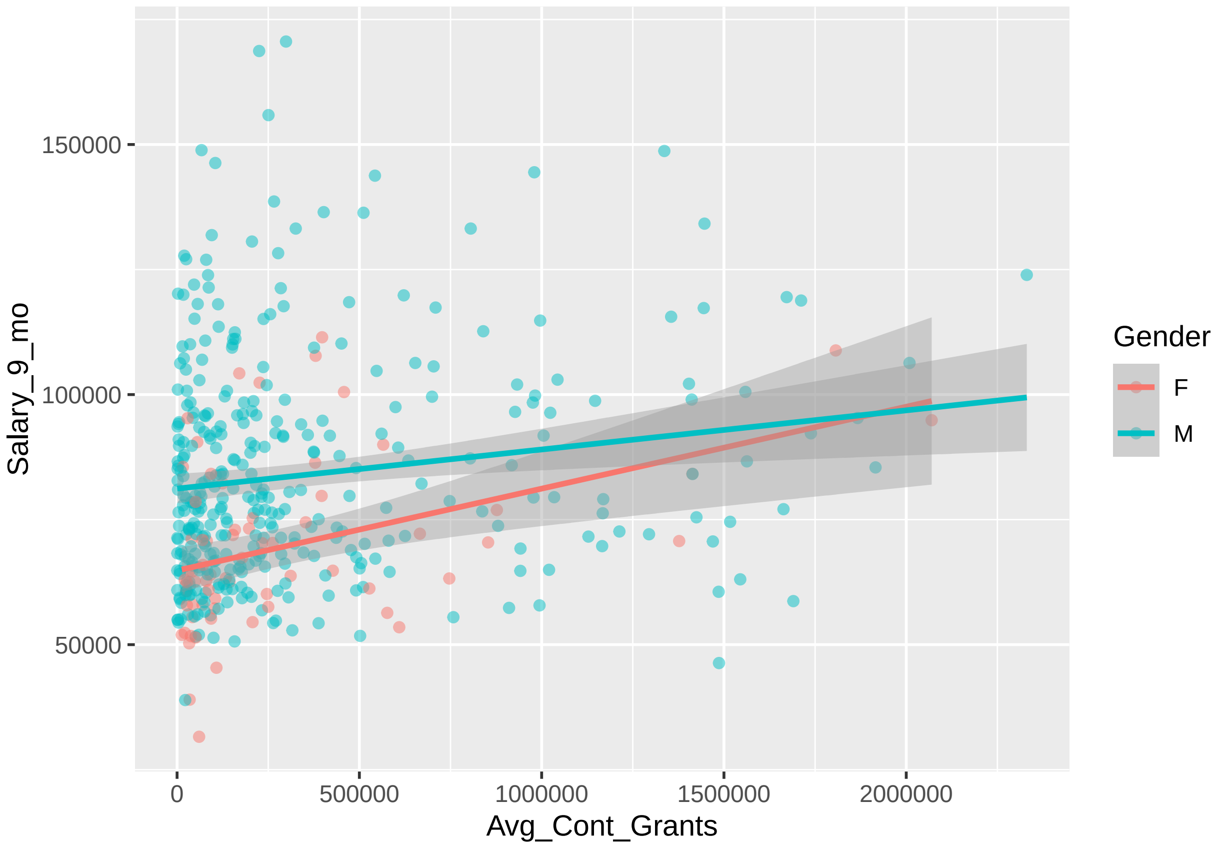
```
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Let's add gender to this picture. Does the relationship we discovered in the first
# part of class apply to grant winning too? We can visualise this relationship using
# ggplot()

ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo, group = Gender)) +
  geom_point(alpha = 0.5, aes(colour = Gender)) +
  geom_smooth(method = "lm", aes(colour = Gender))
```

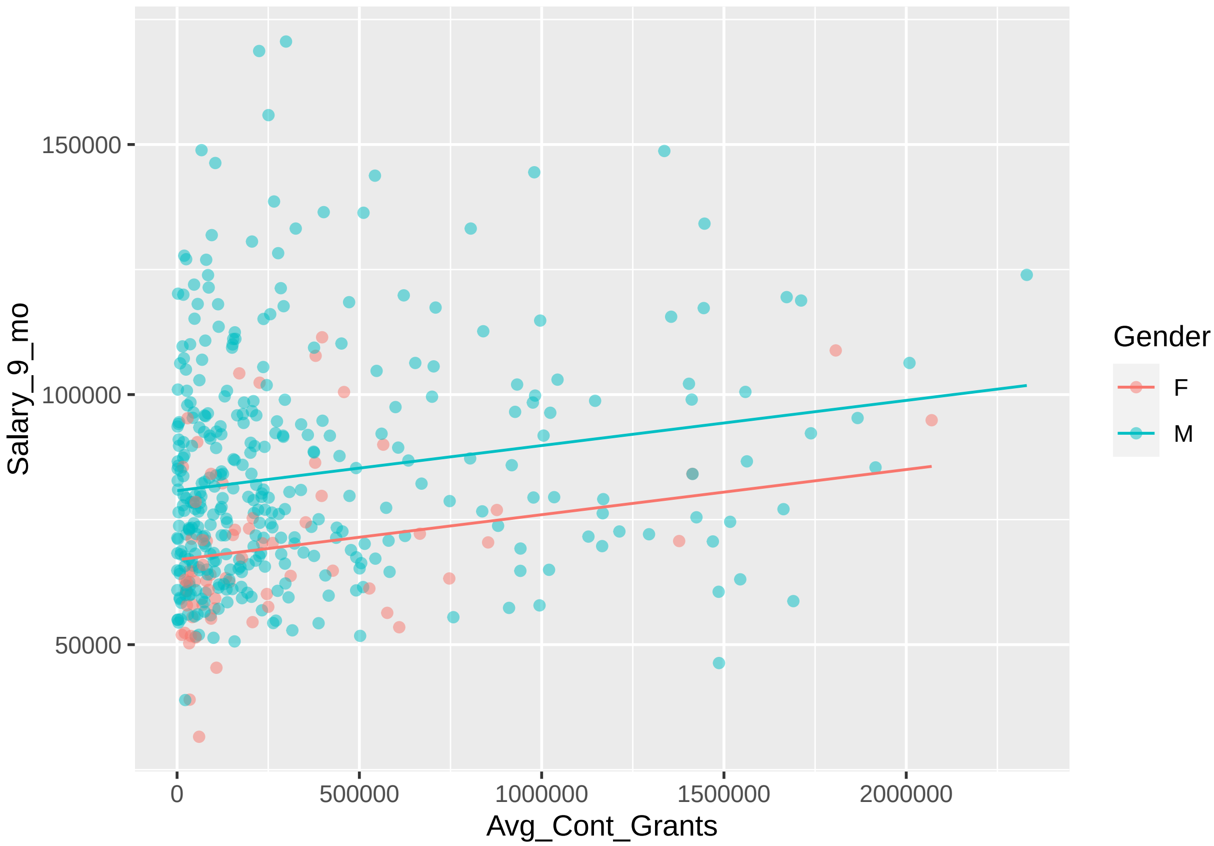
```
## `geom_smooth()` using formula 'y ~ x'
```




```
# How do we interpret this graph?

############################
# Broom and different models
############################

# Thinking about our regression formula, there are two ways of specifying gender in
# our model. We could think of it in terms of a change to our intercept (beta zero),
# or as a change to both our intercept and our slope (beta one). By default, ggplot
# gives us the second of these, which is why our lines have different slopes. What
# if we wanted to model gender just as a change to our intercept? (This is called
# parallel slopes). We need a bit of help from the tidyverse.

## Broom

# The broom package helps us create tidy regression models by *augmenting* our
# datasets with predictions and statistics from our regression model.

mod1 <- lm(Salary_9_mo ~ Avg_Cont_Grants + Gender, data = salary)

# Augment() works like predict(), but creates a data.frame (or tibble) rather than
# a vector

salary_pl <- augment(mod1)

str(salary_pl)
```

```
## tibble [424 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Salary_9_mo    : num [1:424] 87380 58356 59798 64786 54285 ...
##  $ Avg_Cont_Grants: num [1:424] 16667 11667 415871 427231 388192 ...
##  $ Gender         : chr [1:424] "M" "M" "M" "F" ...
##  $ .fitted        : num [1:424] 80934 80889 84536 70818 84287 ...
##  $ .resid         : num [1:424] 6446 -22533 -24739 -6032 -30002 ...
##  $ .hat           : num [1:424] 0.00414 0.00418 0.00287 0.01493 0.00283 ...
##  $ .sigma         : num [1:424] 21420 21394 21388 21420 21372 ...
##  $ .cooksd        : num [1:424] 0.000126 0.00156 0.001287 0.000408 0.001866 ...
##  $ .std.resid     : num [1:424] 0.302 -1.055 -1.158 -0.284 -1.404 ...
##  - attr(*, "terms")=Classes 'terms', 'formula'  language Salary_9_mo ~ Avg_Cont_Grants + Gender
##   .. ..- attr(*, "variables")= language list(Salary_9_mo, Avg_Cont_Grants, Gender)
##   .. ..- attr(*, "factors")= int [1:3, 1:2] 0 1 0 0 0 1
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : chr [1:3] "Salary_9_mo" "Avg_Cont_Grants" "Gender"
##   .. .. .. ..$ : chr [1:2] "Avg_Cont_Grants" "Gender"
##   .. ..- attr(*, "term.labels")= chr [1:2] "Avg_Cont_Grants" "Gender"
##   .. ..- attr(*, "order")= int [1:2] 1 1
##   .. ..- attr(*, "intercept")= int 1
##   .. ..- attr(*, "response")= int 1
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##   .. ..- attr(*, "predvars")= language list(Salary_9_mo, Avg_Cont_Grants, Gender)
##   .. ..- attr(*, "dataClasses")= Named chr [1:3] "numeric" "numeric" "character"
##   .. .. ..- attr(*, "names")= chr [1:3] "Salary_9_mo" "Avg_Cont_Grants" "Gender"
```

```r
# We can now visualise the difference between the *interaction* model and the
# parallel slopes model.

ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo, group = Gender)) +
  geom_point(alpha = 0.5, aes(colour = Gender)) +
  geom_line(data = salary_pl, aes(y = .fitted, colour = Gender)) # we change our data to the fitted val
```

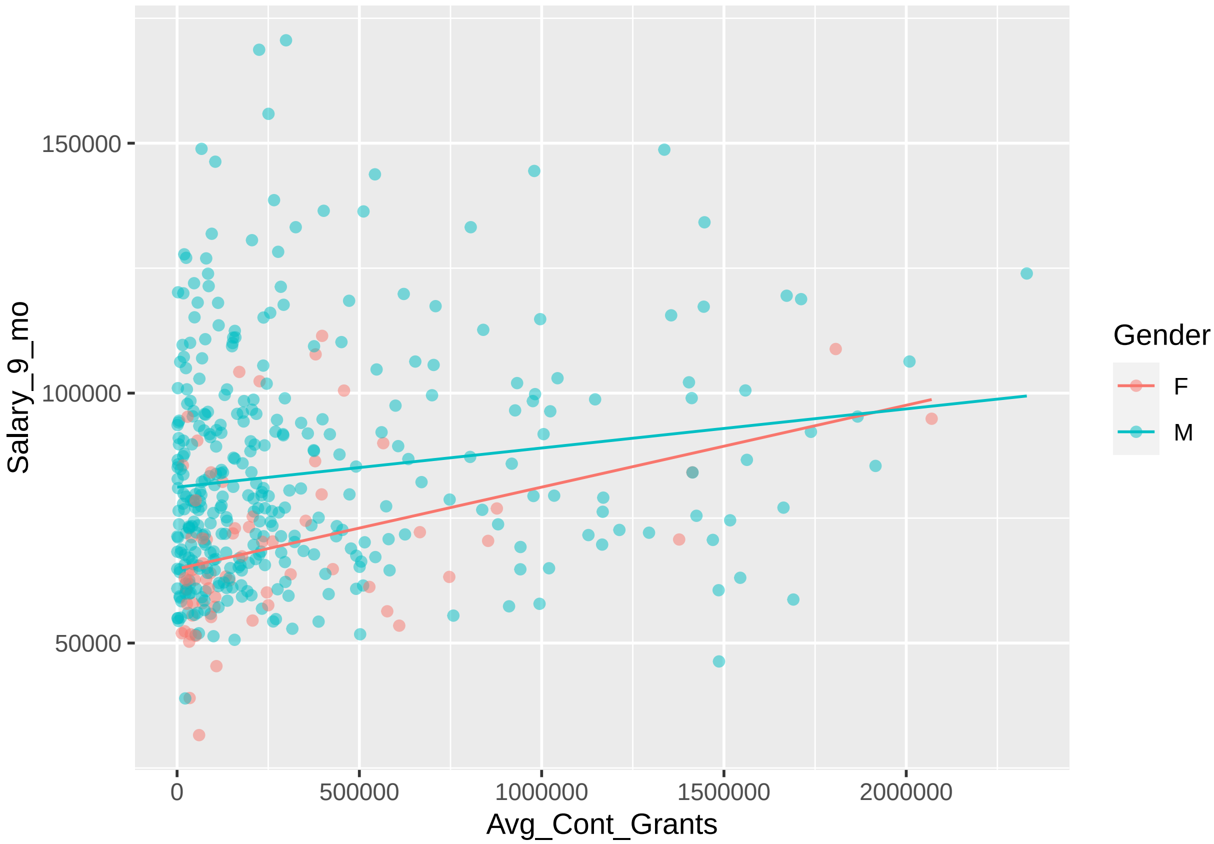



```r
# We can also use augment on to create a data.frame (or tibble) of the *interaction*
# model which ggplot gave us by default

mod2 <- lm(Salary_9_mo ~ Avg_Cont_Grants + Gender + Avg_Cont_Grants:Gender,
           data = salary)

# First, notice the difference in notation: we have to add the interaction between
# grants and gender as a separate term in the equation, separated with a colon.
# Another way of writing this:

lm(Salary_9_mo ~ Avg_Cont_Grants * Gender, data = salary) # gives the same output
```

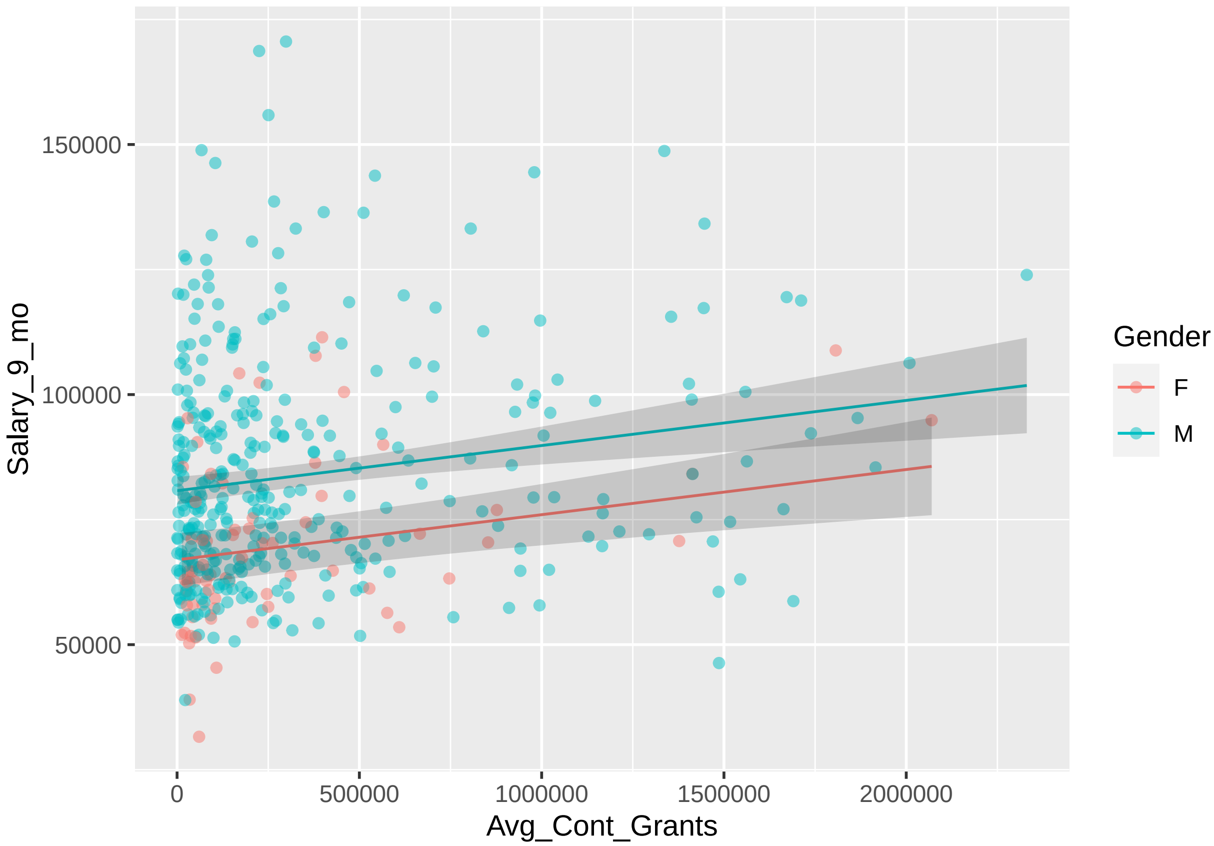
```
##
## Call:
## lm(formula = Salary_9_mo ~ Avg_Cont_Grants * Gender, data = salary)
##
## Coefficients:
##            (Intercept)         Avg_Cont_Grants                 GenderM
##           64810.573323                0.016382            16387.460846
## Avg_Cont_Grants:GenderM
##              -0.008559
```

```r
# We can now augment this second model

salary_int <- augment(mod2)

# And visualise the same way as previously
```

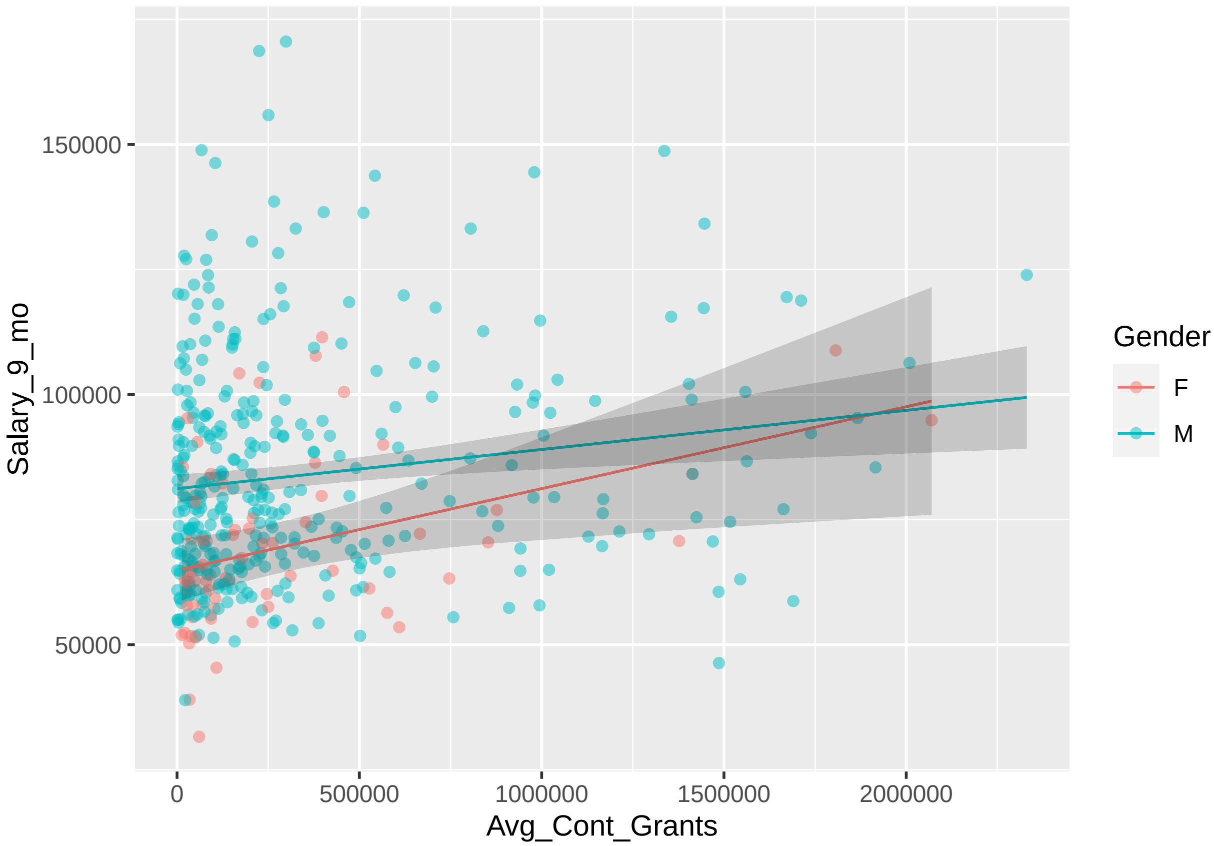
```r
ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo, group = Gender)) +
  geom_point(alpha = 0.5, aes(colour = Gender)) +
  geom_line(data = salary_int, aes(y = .fitted, colour = Gender))
```




```r
# Note: we don't have the error ribbon now, but we could add this using the optional
# interval = "confidence" argument in augment, and the geom_ribbon() function with
# the .lower and .upper columns supplied to the ymin and ymax arguments-

salary_pl <- augment(mod1, interval = "confidence")

ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo, group = Gender)) +
  geom_point(alpha = 0.5, aes(colour = Gender)) +
  geom_line(data = salary_pl, aes(y = .fitted, colour = Gender)) +
  geom_ribbon(data = salary_pl,
              aes(ymin=.lower, ymax=.upper), alpha=0.2)
```

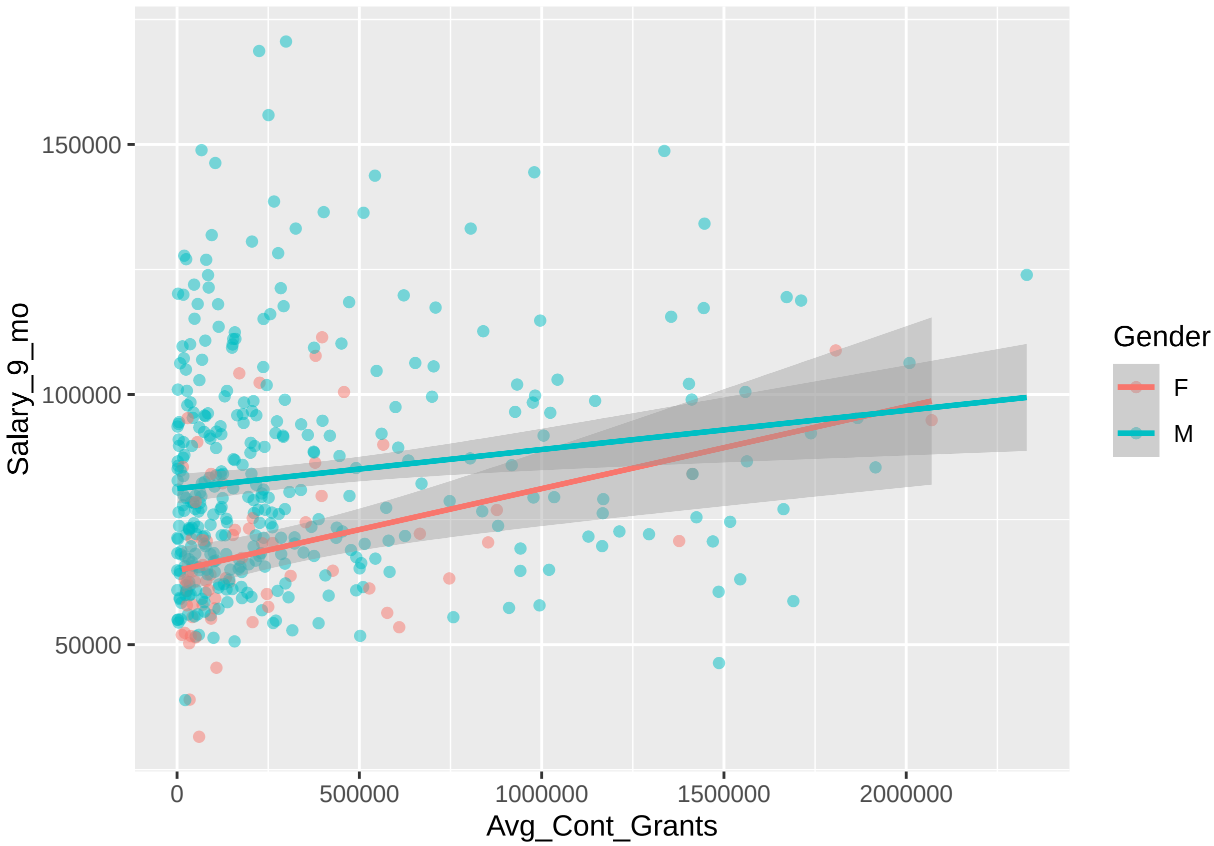



```r
salary_int <- augment(mod2, interval = "confidence")

ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo, group = Gender)) +
  geom_point(alpha = 0.5, aes(colour = Gender)) +
  geom_line(data = salary_int, aes(y = .fitted, colour = Gender)) +
  geom_ribbon(data = salary_int,
              aes(ymin=.lower, ymax=.upper), alpha=0.2)
```




```
# Note: our error bars for gender = female are actually a bit wider this way than
# using ggplot. This is likely due to a difference in weighting for extreme data.

ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo, group = Gender)) +
  geom_point(alpha = 0.5, aes(colour = Gender)) +
  geom_smooth(method = "lm", aes(colour = Gender))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```r
#######################
# Which model is better?
#######################

summary(mod1)
```

```
##
## Call:
## lm(formula = Salary_9_mo ~ Avg_Cont_Grants + Gender, data = salary)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47876 -15358  -4189  11097  87111
##
## Coefficients:
##                     Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)     66962.548901  2686.473019  24.926 < 0.0000000000000002 ***
## Avg_Cont_Grants     0.009025     0.002379   3.793             0.000171 ***
## GenderM         13820.729492  2834.513239   4.876           0.00000154 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21400 on 421 degrees of freedom
## Multiple R-squared:  0.08652,    Adjusted R-squared:  0.08218
## F-statistic: 19.94 on 2 and 421 DF,  p-value: 0.00000000534
```

```r
summary(mod2)
```

```
##
## Call:
## lm(formula = Salary_9_mo ~ Avg_Cont_Grants + Gender + Avg_Cont_Grants:Gender,
##     data = salary)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -46504 -15371  -3797  11015  87055
##
## Coefficients:
##                            Estimate   Std. Error t value             Pr(>|t|)
## (Intercept)            64810.573323  3188.927084  20.324 < 0.0000000000000002
## Avg_Cont_Grants            0.016382     0.006346   2.581               0.0102
## GenderM                16387.460846  3498.166184   4.685           0.00000379
## Avg_Cont_Grants:GenderM   -0.008559     0.006845  -1.250               0.2118
##
## (Intercept)            ***
## Avg_Cont_Grants        *
## GenderM                ***
## Avg_Cont_Grants:GenderM
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21380 on 420 degrees of freedom
## Multiple R-squared:  0.0899, Adjusted R-squared:  0.0834
## F-statistic: 13.83 on 3 and 420 DF,  p-value: 0.00000001287
```

```r
# What do you think? Compare the outputs above, and pay close attention to the
# significance of the different coefficients in each model.

##########
# Exercise
##########

# Try the same approach using Rank_Code as a term in your regression analysis. What
# do you find? How does the regression model change when we substitute grant and
# gender into this model?

salary %>%
  group_by(Rank_Code) %>%
  summarise(mean = mean(Salary_9_mo))
```

```
## # A tibble: 3 x 2
##   Rank_Code   mean
##       <dbl>  <dbl>
## 1         1 94177.
## 2         2 71205.
## 3         3 62539.
```

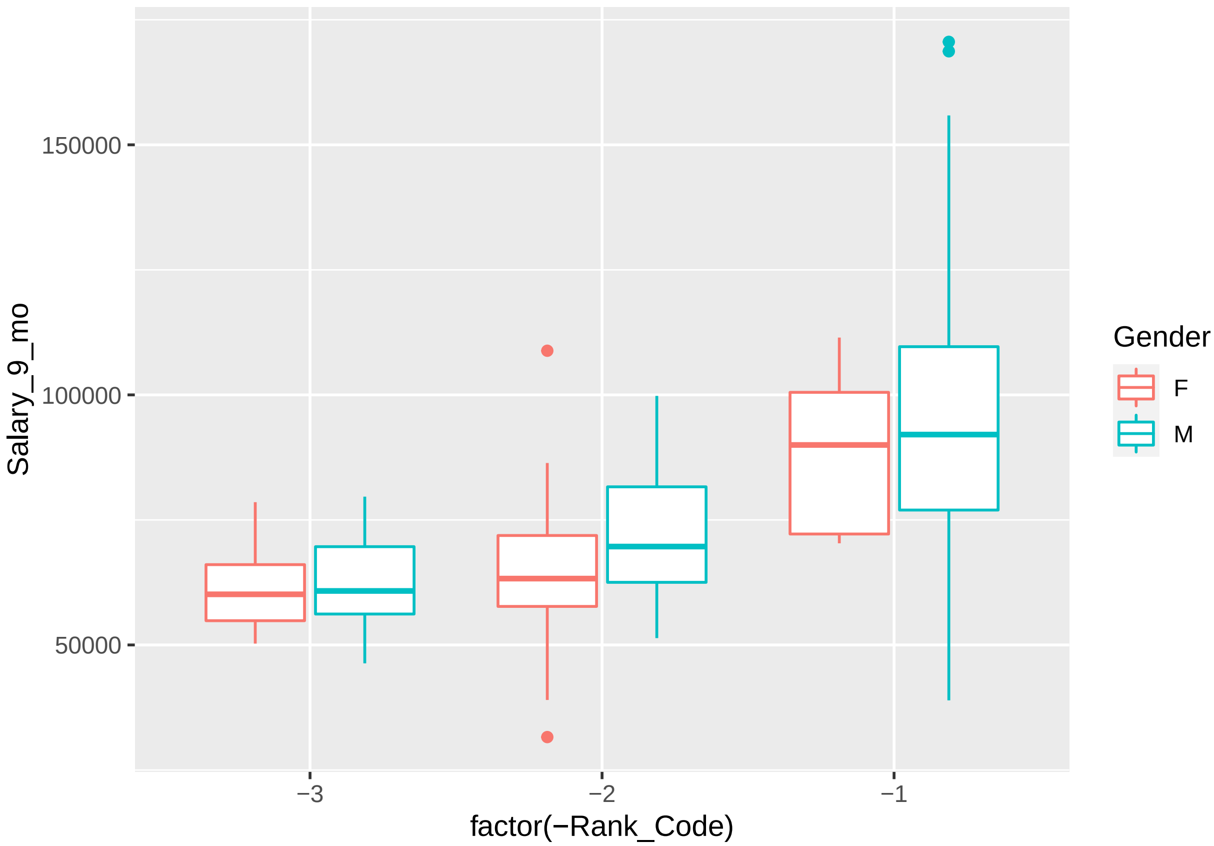
```r
# It seems that Rank_Code is a variable referring to seniority. Let's see how this
# relates to gender

salary %>%
```

```r
  group_by(Rank_Code, Gender) %>%
  summarise(mean = mean(Salary_9_mo)) %>%
  pivot_wider(names_from = Gender, values_from = mean) %>% # reshape to make a table
  ungroup() %>% # ungroup to be able to add a column
  mutate(diff = M - F) # add a new column with the difference in means
```

```
## `summarise()` has grouped output by 'Rank_Code'. You can override using the `.groups` argument.
```

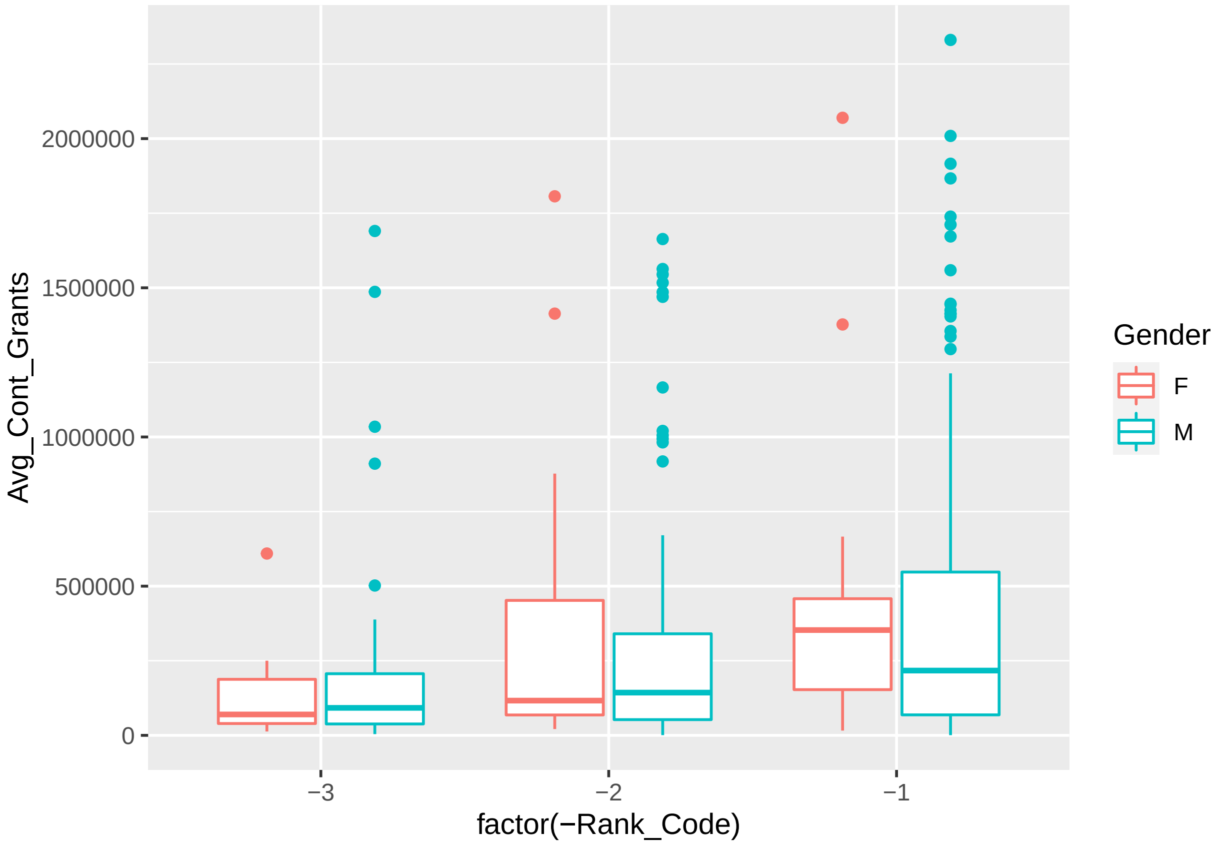
```
## # A tibble: 3 x 4
##   Rank_Code      F      M  diff
##       <dbl>  <dbl>  <dbl> <dbl>
## 1         1 87825. 94714. 6889.
## 2         2 65378. 72701. 7323.
## 3         3 61276. 63170. 1894.
```

```r
# From a quick manipulation of the data, it looks like the gender effect is present
# across all grades, but is greatest in the middle grade.

ggplot(salary, aes(factor(-Rank_Code), Salary_9_mo)) +
  geom_boxplot(aes(colour = Gender))
```

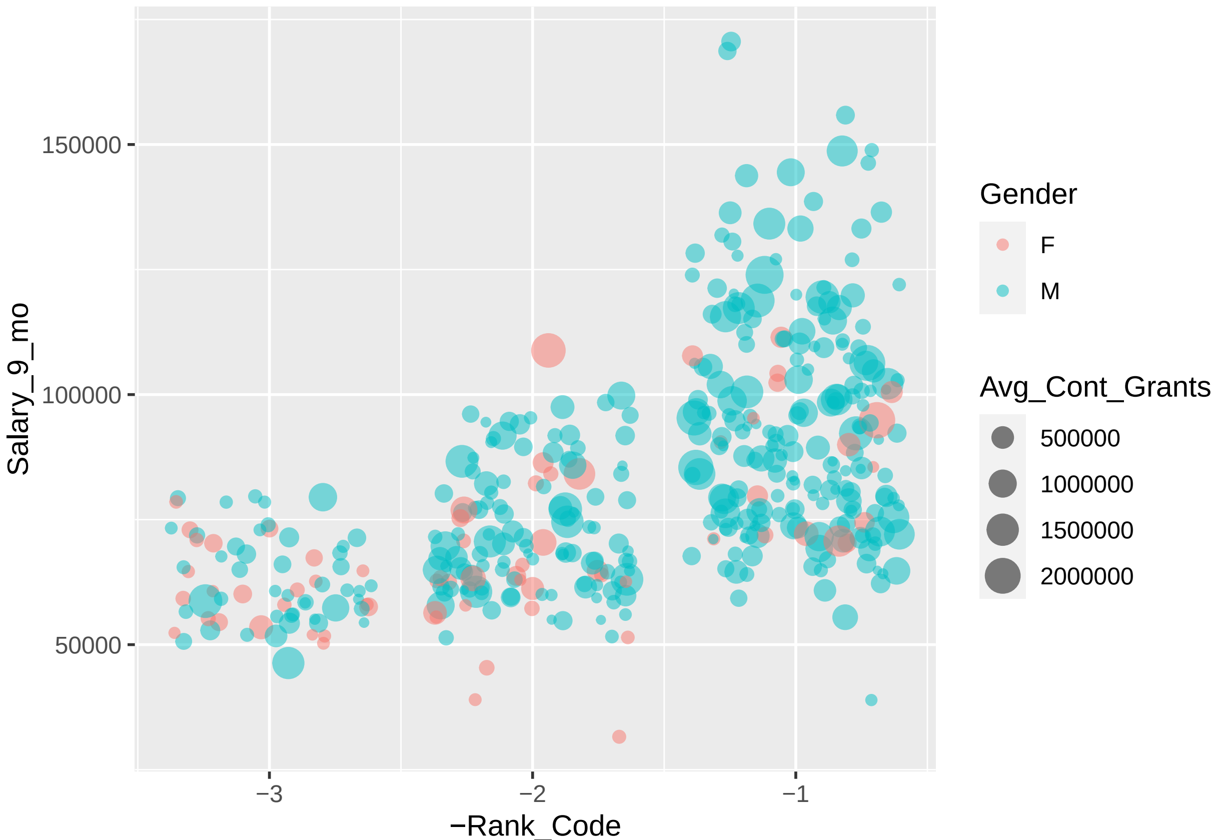



```r
# The boxplot shows this quite nicely: the medians are closely grouped for 3 and 1,
# but there is a bigger gender gap in rank 2.

# Rather than a linear regression, when our output variable is continuous and our
# predictors are categorical, we often use ANOVA. The difference in terms of outputs
# however is small, as ANOVA and regression are both part of the general linear model.
```

```r
# Perhaps we're thinking about this wrong. Maybe men get paid more because they bring
# in more grant money... And maybe people are ranked according to how much money they
# bring in. We can quickly check this by substituting salary for grants

ggplot(salary, aes(factor(-Rank_Code), Avg_Cont_Grants)) +
  geom_boxplot(aes(colour = Gender))
```




```r
# Oh, fancy that. It seems that the median grant contribution for women in rank 1 is
# actually higher than their male counterparts. On the other hand, there are a lot
# of outliers. Plus, there is a chance that we have a correlation between grant
# contribution and salary, which interacts with gender and rank. Let's see if we can
# visualise all those things together to make sense of the relationships.

ggplot(salary, aes(-Rank_Code, # we use "-" here because rank is in reverse order
                   Salary_9_mo,
                   colour = Gender,
                   size = Avg_Cont_Grants)) +
  geom_jitter(alpha = 0.5) #a good use case for jittering!
```




```
# After a bit of experimentation, I think this is the most instructive visualisation.
# Firstly, we see how many more men there are in this dataset. Secondly, although the
# data are a bit noisy, we can see that there is an underlying trend within each rank
# of those with bigger grants getting paid more. Women seem to conform to this trend,
# with the possible exception of rank 1, where there seems to be an upper segment of
# male academics whose salaries are way above any woman's, and indeed their male peers.
# Let's run a regression and see the precise relationship between all these variables.

mod3 <- lm(data = salary, Salary_9_mo ~ Rank_Code + Gender + Avg_Cont_Grants)
summary(mod3)
```

```
##
## Call:
## lm(formula = Salary_9_mo ~ Rank_Code + Gender + Avg_Cont_Grants,
##     data = salary)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52668 -13012  -1845  10982  77801
##
## Coefficients:
##                     Estimate    Std. Error t value            Pr(>|t|)
## (Intercept)     101964.083045   3531.514211  28.873 <0.0000000000000002 ***
## Rank_Code       -16104.482307   1243.331762 -12.953 <0.0000000000000002 ***
## GenderM           5634.861173   2480.766234   2.271              0.0236 *
## Avg_Cont_Grants      0.004336      0.002046   2.119              0.0347 *
## ---
```

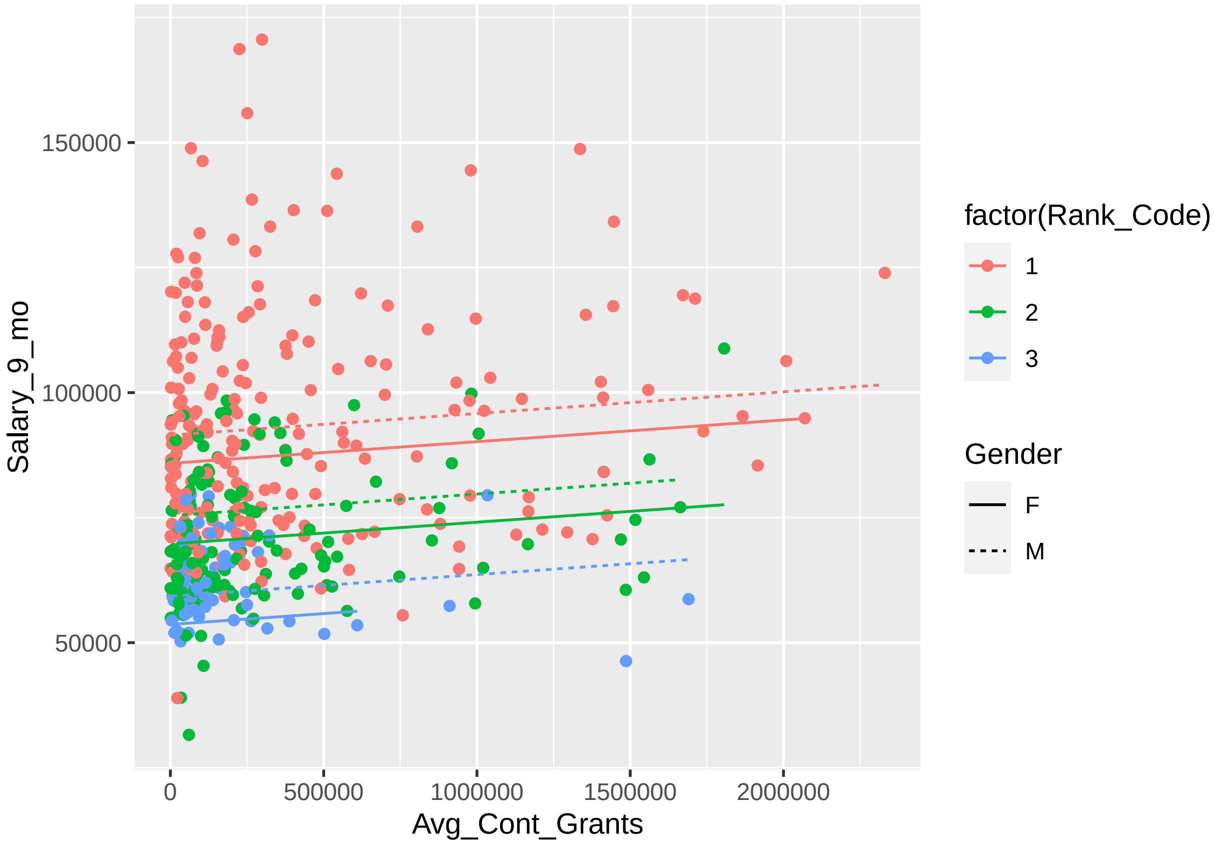
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18110 on 420 degrees of freedom
## Multiple R-squared:  0.3473, Adjusted R-squared:  0.3426
## F-statistic: 74.48 on 3 and 420 DF,  p-value: < 0.00000000000000022
```

```
# Well, we can see from our first attempt at modelling these variables that rank is
# a big association, which is also statistically significant. With this variable
# included, gender loses some statistical significance (though it remains a big
# association, and we see from the standard error that it is way above zero). The
# role of grants seems now to be less important, though we obviously need to scale
# the coefficient appropriately, i.e. an extra $50,000 grant is associated with an
# extra $200 in salary...

# Let's try a couple more models: one where we interact gender with rank, and one
# where we drop the grants terms.

mod4 <- lm(data = salary, Salary_9_mo ~ Rank_Code * Gender + Avg_Cont_Grants)
summary(mod4)
```

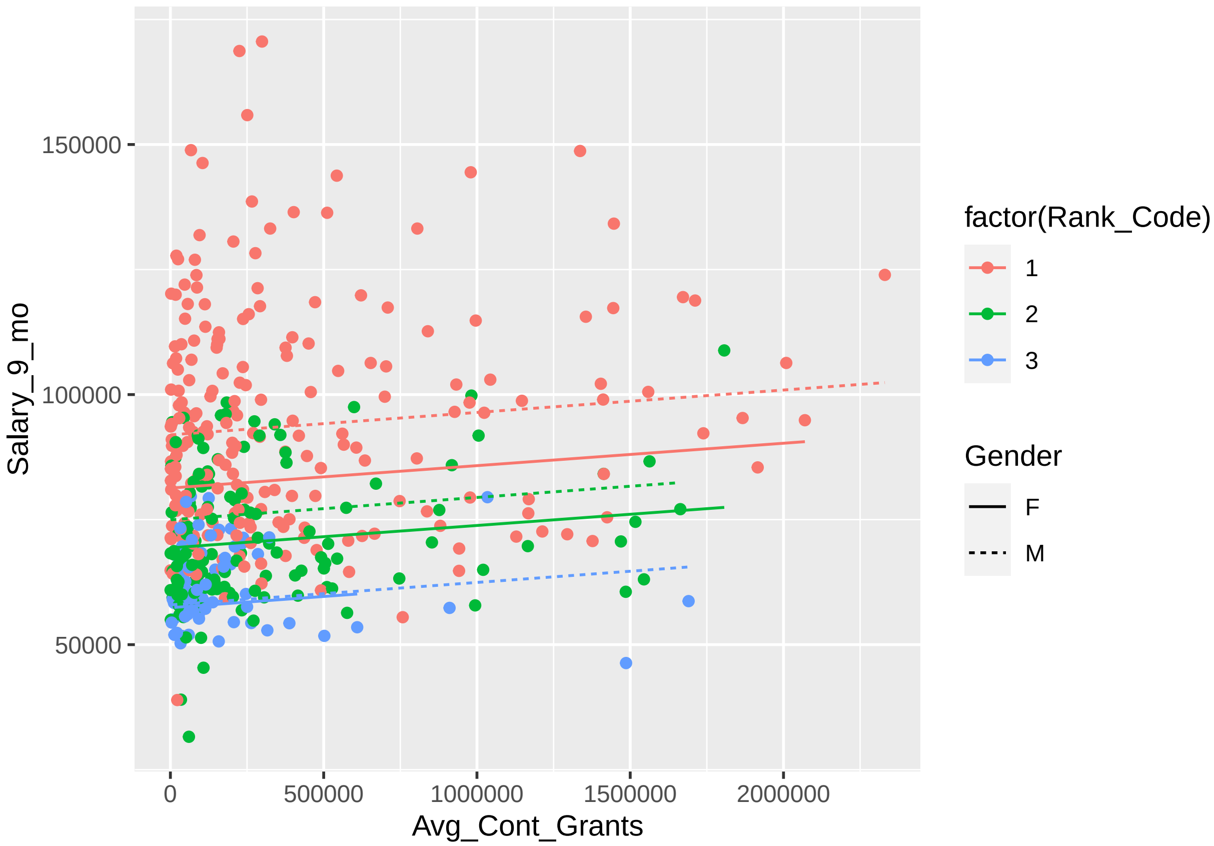
```
##
## Call:
## lm(formula = Salary_9_mo ~ Rank_Code * Gender + Avg_Cont_Grants,
##     data = salary)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -53124 -12564  -2254  10971  77304
##
## Coefficients:
##                        Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)        93267.070586   6527.068464  14.289 < 0.0000000000000002 ***
## Rank_Code         -11960.189566   2896.953212  -4.129            0.0000441 ***
## GenderM            15677.136306   6809.183982   2.302               0.0218 *
## Avg_Cont_Grants        0.004482      0.002044   2.192               0.0289 *
## Rank_Code:GenderM  -5036.346230   3181.071986  -1.583               0.1141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18080 on 419 degrees of freedom
## Multiple R-squared:  0.3511, Adjusted R-squared:  0.3449
## F-statistic: 56.69 on 4 and 419 DF,  p-value: < 0.00000000000000022
```

```
# This is an interesting development: we see that the statistical significance
# doesn't change very much, neither do we find any statistical significance for our
# interaction term. Look at the change in the estimates though: being male is now
# associated with an extra $15677 salary, versus $5634 last time. This is significant
# at roughly the same level as our previous model. How to interpret this? Let's do
# some visualisations of the two models.

# Model 3: No interaction
ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo)) +
  geom_point(aes(colour = factor(Rank_Code))) +
  geom_line(data = augment(mod3), aes(y = .fitted,
                                      linetype = Gender,
```

```
                                   colour = factor(Rank_Code)))
```




```
# Model 4: Interaction
ggplot(salary, aes(Avg_Cont_Grants, Salary_9_mo)) +
  geom_point(aes(colour = factor(Rank_Code))) +
  geom_line(data = augment(mod4), aes(y = .fitted,
                                   linetype = Gender,
                                   colour = factor(Rank_Code)))
```




```
# What changed? By visualising the two models, we can see that in the first model
# (no interaction) the distance between gender is held constant at each level of
# rank. In the second model, the distance is allowed to vary within the different
# levels. Think about an average of the three "rank" lines for each gender in each
# plot: in the first plot, the female average would be higher up on the y axis, hence
# less of a difference between male and female in the regression coefficients).In the
# second plot, the female average is lower down, and further away from the male,
# hence the bigger association with gender. Which is "better"? Because there's such
# a lot of noise in the plot, all we can really go off is the t score for each model:
# because this is slightly improved for gender on the second model, we might favour
# that one, whilst noting that the interaction term itself is not significant (which
# is to say, the association of gender *within* rank is very noisy, as we see from
# the visualisation.) Note: both of these models are parallel slopes, because we were
# interacting two categorical variables with each other, not with a continuous
# variable (as we did in the class exercise). We could try interacting all of these
# with the grants variable, but we're then creating a lot of interaction terms, and
# this can become hard to interpret.

# Model 5: dropping grants

mod5 <- lm(mod4 <- lm(data = salary, Salary_9_mo ~ Rank_Code * Gender))
summary(mod5)
```

```
##
## Call:
## lm(formula = mod4 <- lm(data = salary, Salary_9_mo ~ Rank_Code *
##     Gender))
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -54811 -12758  -1855  11312  76856
##
## Coefficients:
##                  Estimate Std. Error t value            Pr(>|t|)
## (Intercept)         96123       6425  14.962 < 0.0000000000000002 ***
## Rank_Code          -12700       2890  -4.394           0.0000141 ***
## GenderM             15033       6834   2.200              0.0284 *
## Rank_Code:GenderM   -4721       3192  -1.479              0.1399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18160 on 420 degrees of freedom
## Multiple R-squared:  0.3437, Adjusted R-squared:  0.339
## F-statistic: 73.32 on 3 and 420 DF,  p-value: < 0.00000000000000022
```

```r
# As we see from our summary, dropping grants reduces the statistical significance of
# our other predictors. We might therefore be encouraged to leave it in, even though
# it is only significant at the 0.05 level.
```