DSA 8020 R Session 5: Multiple Linear Regression IV

Whitney

January 31, 2021

Contents

Regression with Both Quantitative and Qualitative Predictors

Salaries for Professors Data Set

The 2008-09 nine-month academic salary for Assistant Professors, Associate Professors and Professors in a college in the U.S. The data were collected as part of the on-going effort of the college's administration to monitor salary differences between male and female faculty members.

Load the data

```r
library(carData)
data(Salaries)
head(Salaries)
```

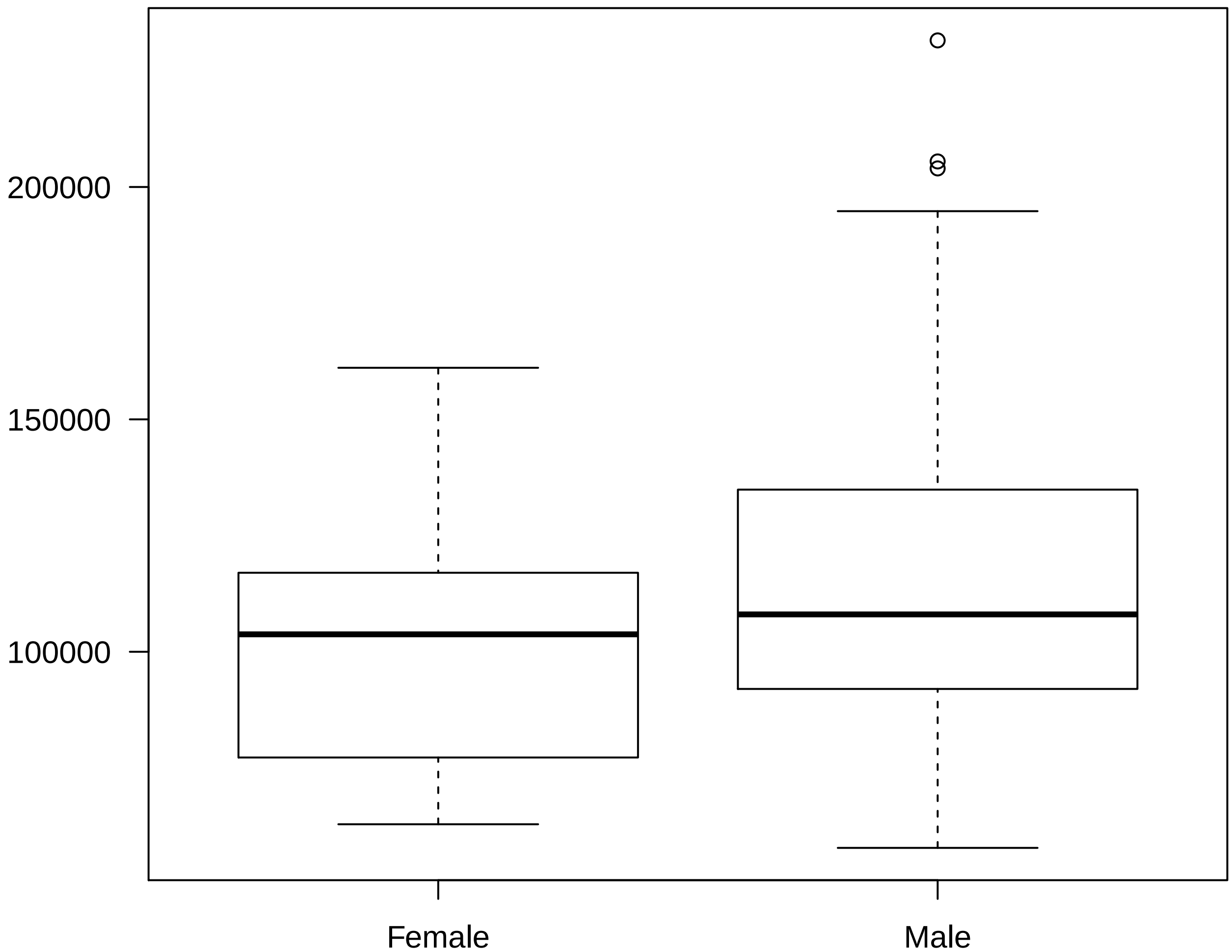
```
##        rank discipline yrs.since.phd yrs.service  sex salary
## 1      Prof          B            19          18 Male 139750
## 2      Prof          B            20          16 Male 173200
## 3  AsstProf          B             4           3 Male  79750
## 4      Prof          B            45          39 Male 115000
## 5      Prof          B            40          41 Male 141500
## 6 AssocProf          B             6           6 Male  97000
```

Summazrize the data

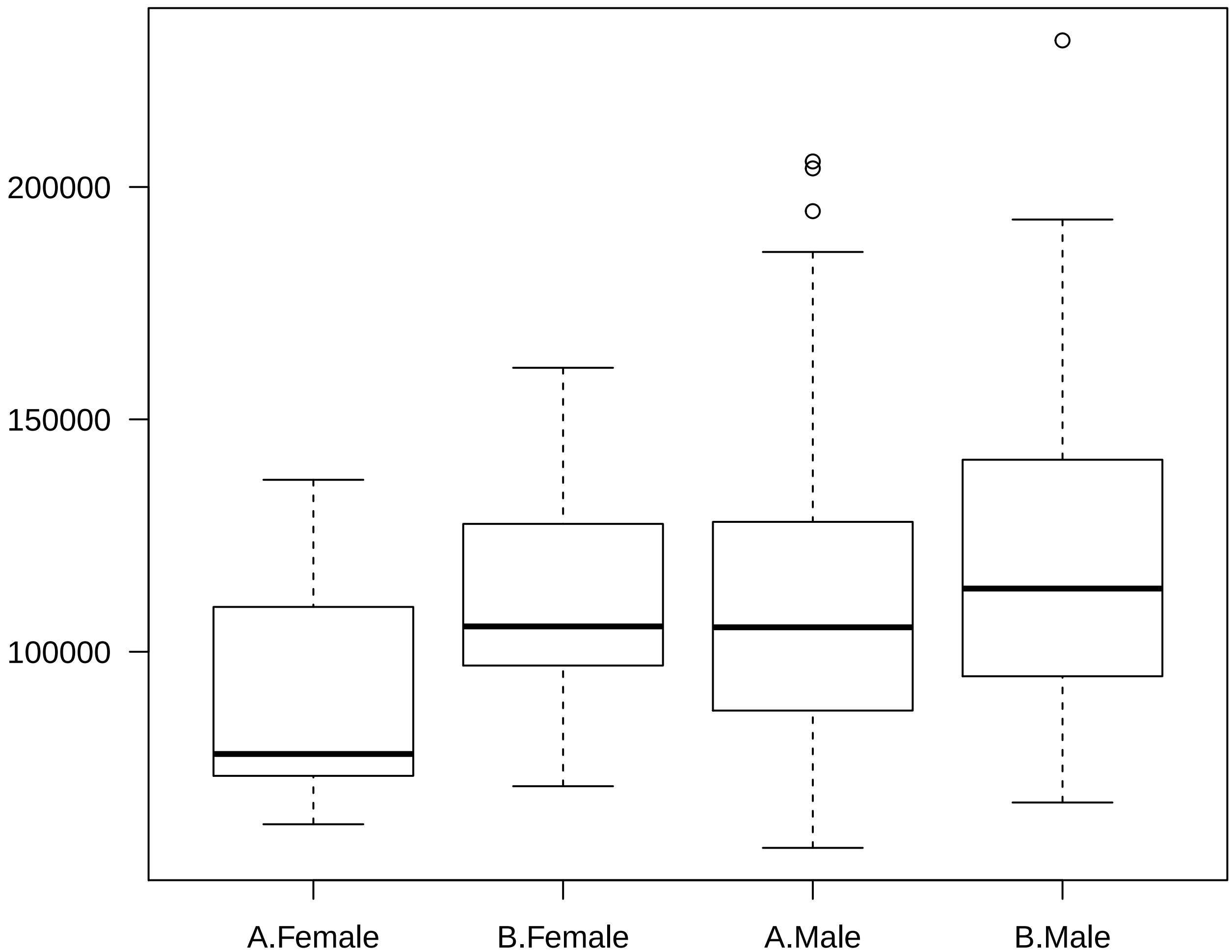
```
summary(Salaries)
```

```
##         rank     discipline yrs.since.phd    yrs.service        sex
##  AsstProf : 67   A:181      Min.   : 1.00   Min.   : 0.00   Female: 39
##  AssocProf: 64   B:216      1st Qu.:12.00   1st Qu.: 7.00   Male  :358
##  Prof     :266              Median :21.00   Median :16.00
##                             Mean   :22.31   Mean   :17.61
##                             3rd Qu.:32.00   3rd Qu.:27.00
##                             Max.   :56.00   Max.   :60.00
##      salary
##  Min.   : 57800
##  1st Qu.: 91000
##  Median :107300
##  Mean   :113706
##  3rd Qu.:134185
##  Max.   :231545
```

```
boxplot(salary ~ sex, data = Salaries, las = 1)
```



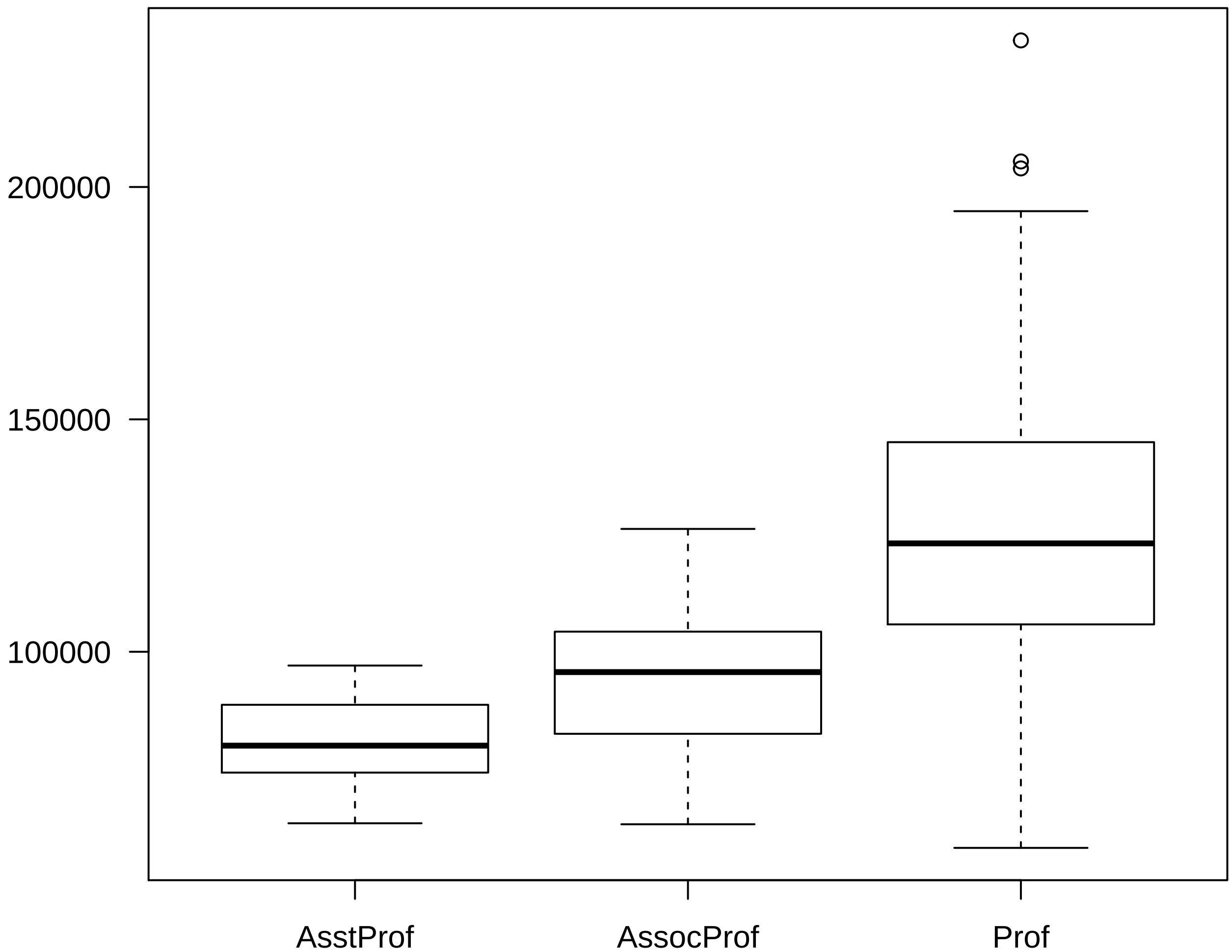

```
boxplot(salary ~ discipline + sex, data = Salaries, las = 1)
```



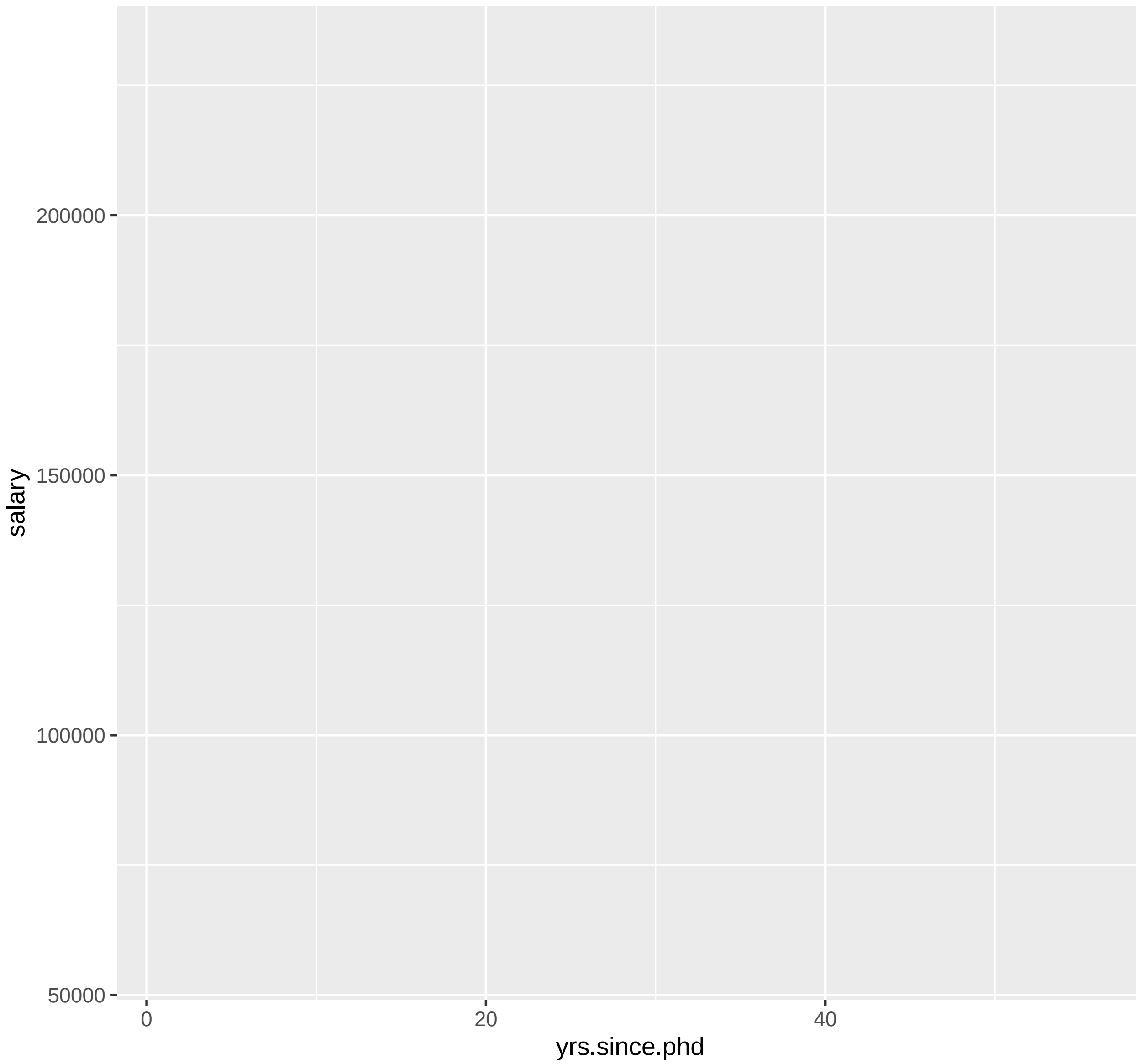

```r
boxplot(salary ~ rank, data = Salaries, las = 1)
# Cross tabulation
xtabs(~ sex + rank + discipline, data = Salaries)
```

```
## , , discipline = A
##
##         rank
## sex      AsstProf AssocProf Prof
##   Female        6         4    8
##   Male         18        22  123
##
## , , discipline = B
##
##         rank
## sex      AsstProf AssocProf Prof
##   Female        5         6   10
##   Male         38        32  125
```

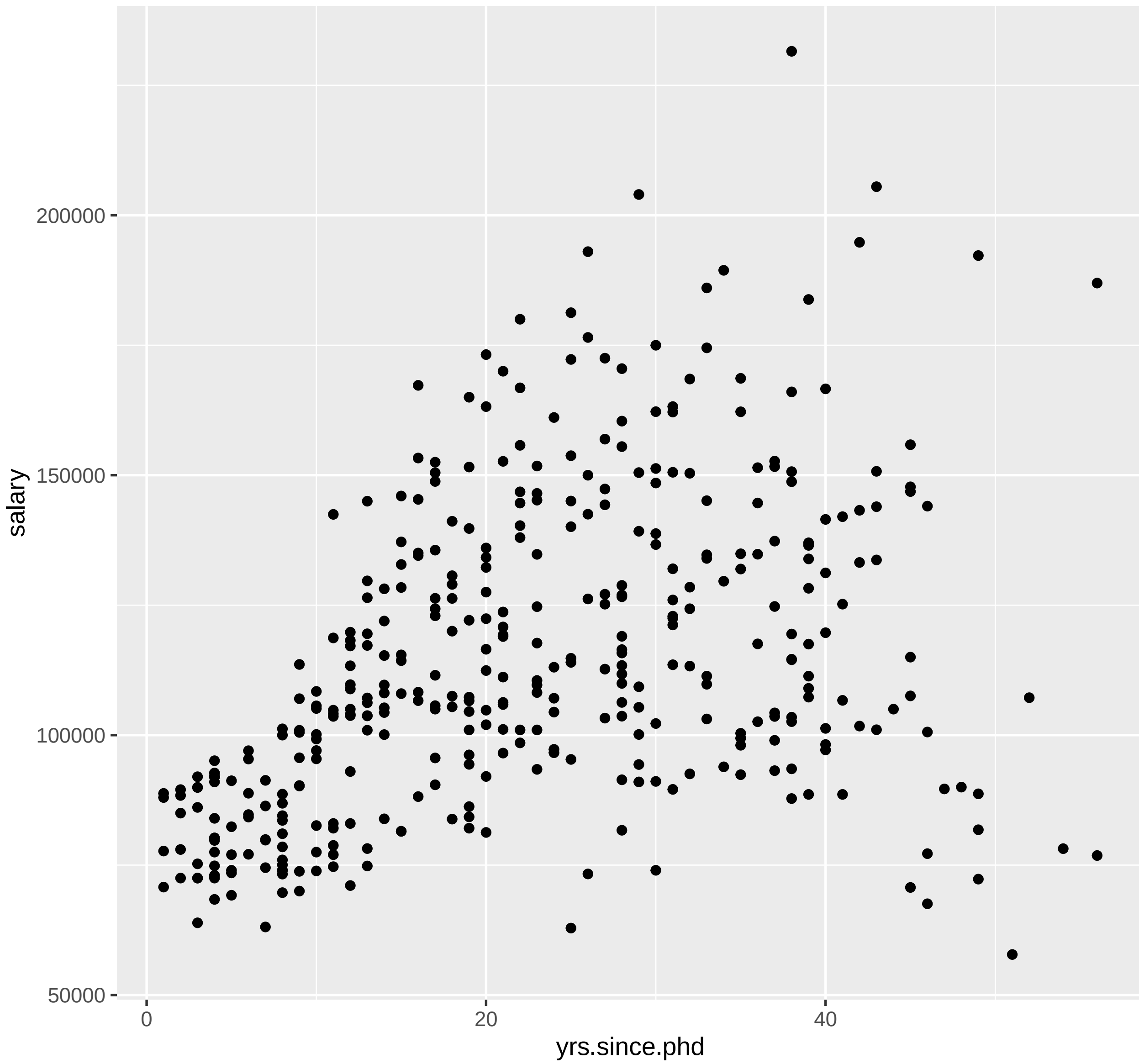
```r
# Plot salary vs. yrs since Ph.D. by gender using ggplot
library(ggplot2)
```



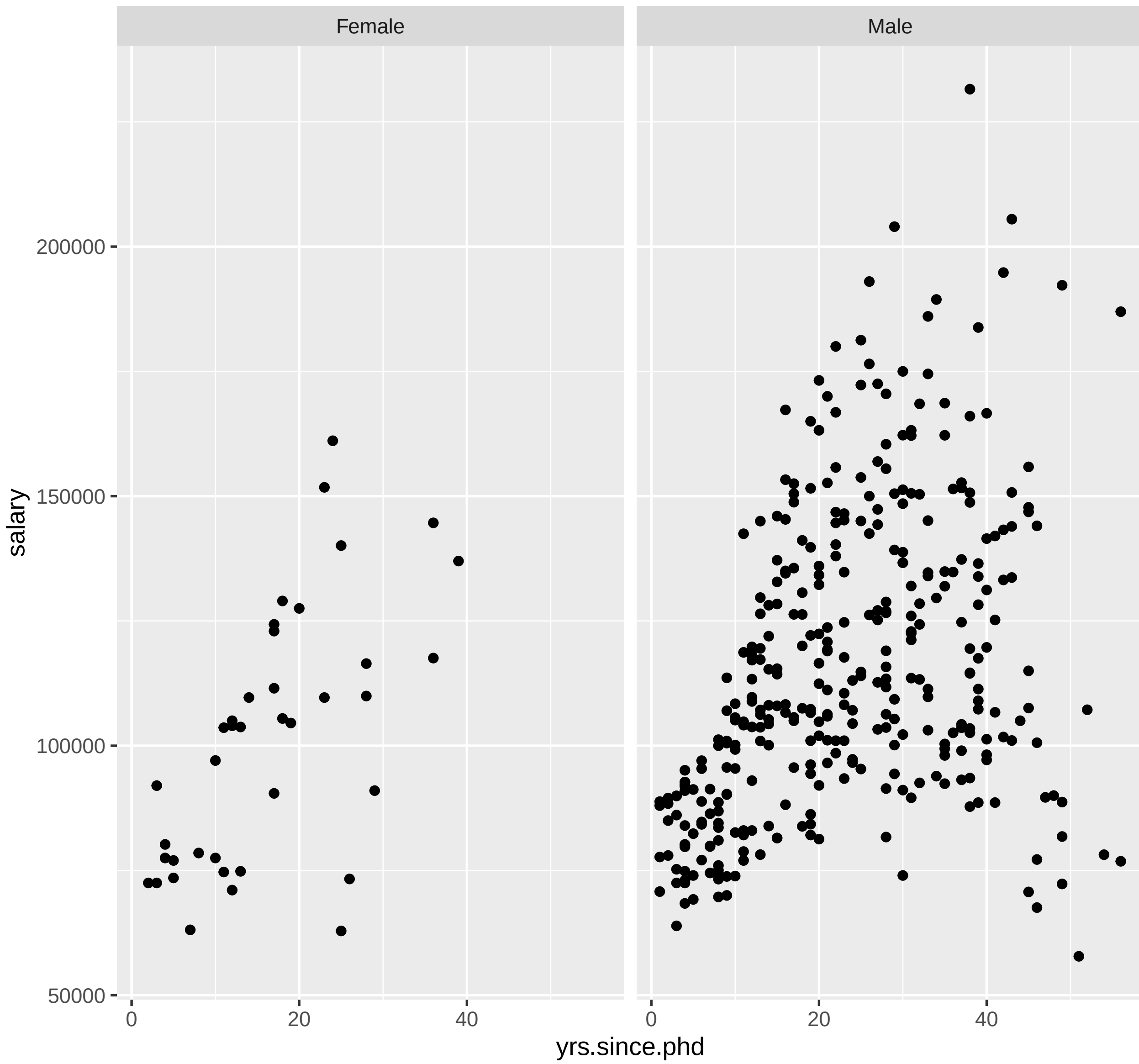

```
(plot1 <- ggplot(aes(x = yrs.since.phd, y = salary), data = Salaries))
```




```r
(plot2 <- plot1 + geom_point())
```

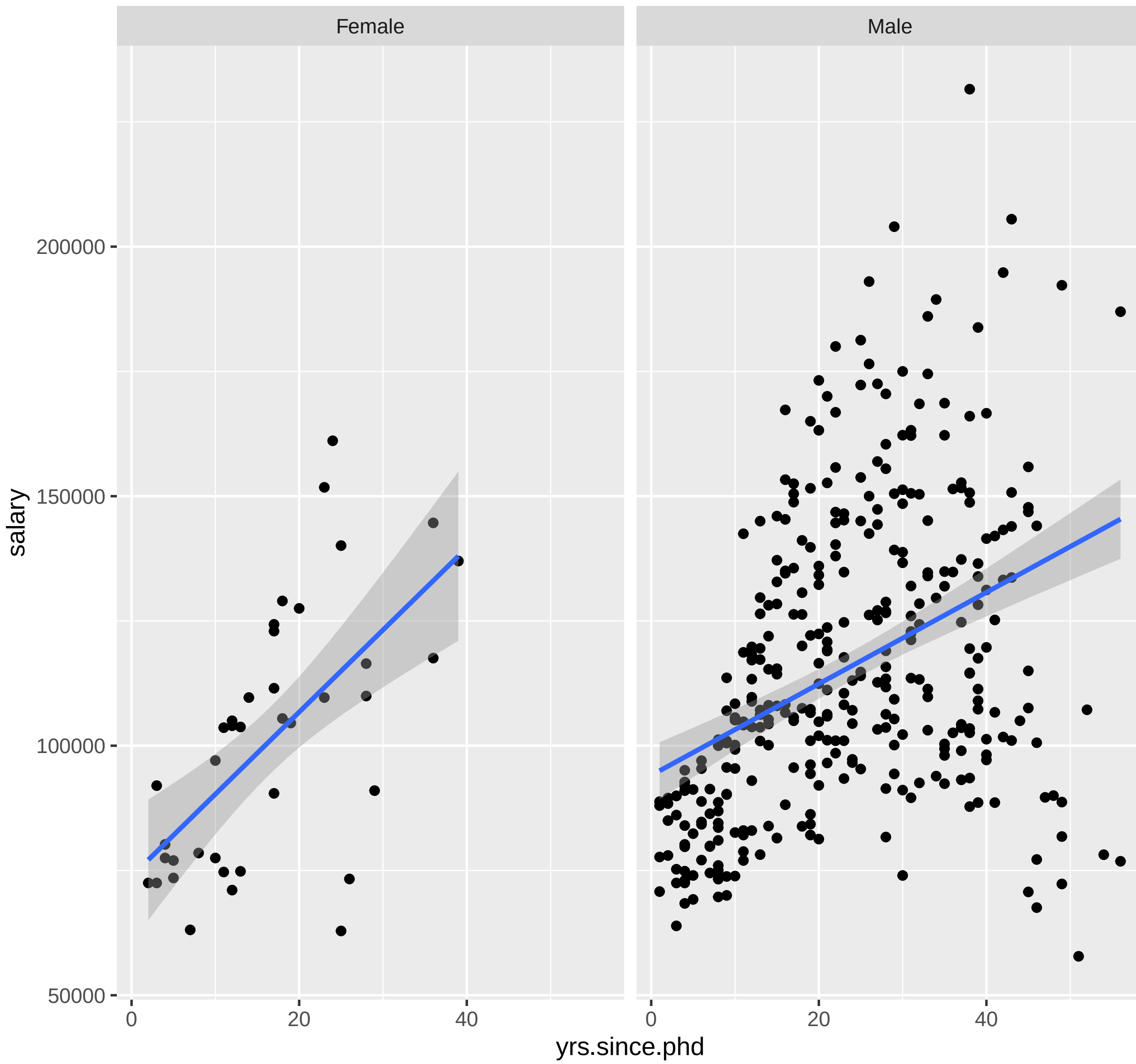



```
(plot3 <- plot2 + facet_grid(~ sex))
```




```
(plot4 <- plot3 + geom_smooth(method = "lm"))
```

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



Model fitting

Model 1: A MLR with `yrs.since.phd` (numerical predictor), discipline, rank, and sex (categorical predictors)

```
m1 <- lm(salary ~ discipline + rank + sex + yrs.since.phd, data = Salaries)
X <- model.matrix(m1)
head(X)
```

```
##   (Intercept) disciplineB rankAssocProf rankProf sexMale yrs.since.phd
## 1           1           1             0        1       1            19
## 2           1           1             0        1       1            20
## 3           1           1             0        0       1             4
## 4           1           1             0        1       1            45
## 5           1           1             0        1       1            40
## 6           1           1             1        0       1             6
```

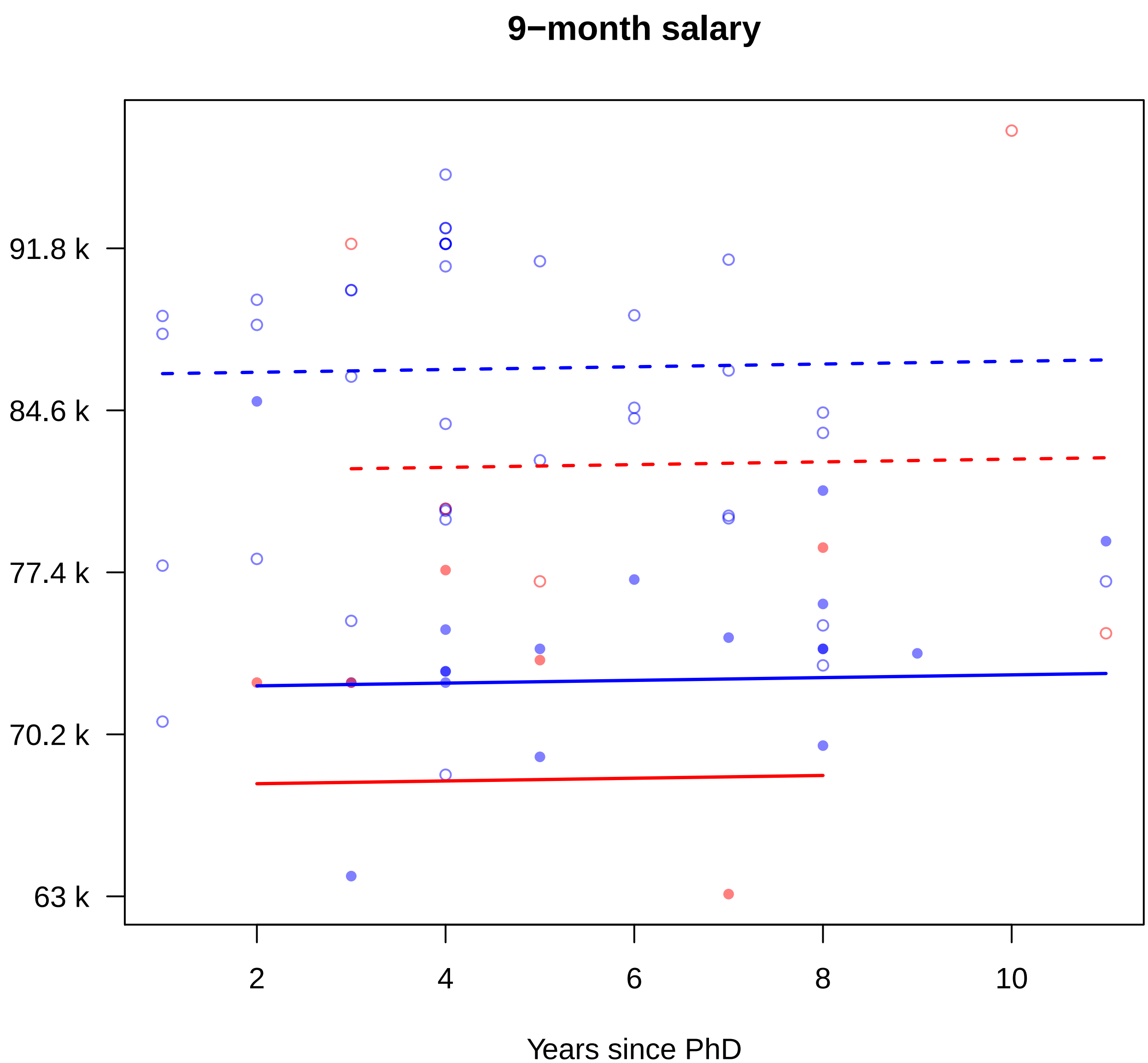
```r
summary(m1)
```

```
##
## Call:
## lm(formula = salary ~ discipline + rank + sex + yrs.since.phd,
##     data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -67451 -13860  -1549  10716  97023
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   67884.32    4536.89  14.963  < 2e-16 ***
## disciplineB   13937.47    2346.53   5.940 6.32e-09 ***
## rankAssocProf 13104.15    4167.31   3.145  0.00179 **
## rankProf      46032.55    4240.12  10.856  < 2e-16 ***
## sexMale        4349.37    3875.39   1.122  0.26242
## yrs.since.phd    61.01     127.01   0.480  0.63124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22660 on 391 degrees of freedom
## Multiple R-squared:  0.4472, Adjusted R-squared:  0.4401
## F-statistic: 63.27 on 5 and 391 DF,  p-value: < 2.2e-16
```

Plot the model 1 fit

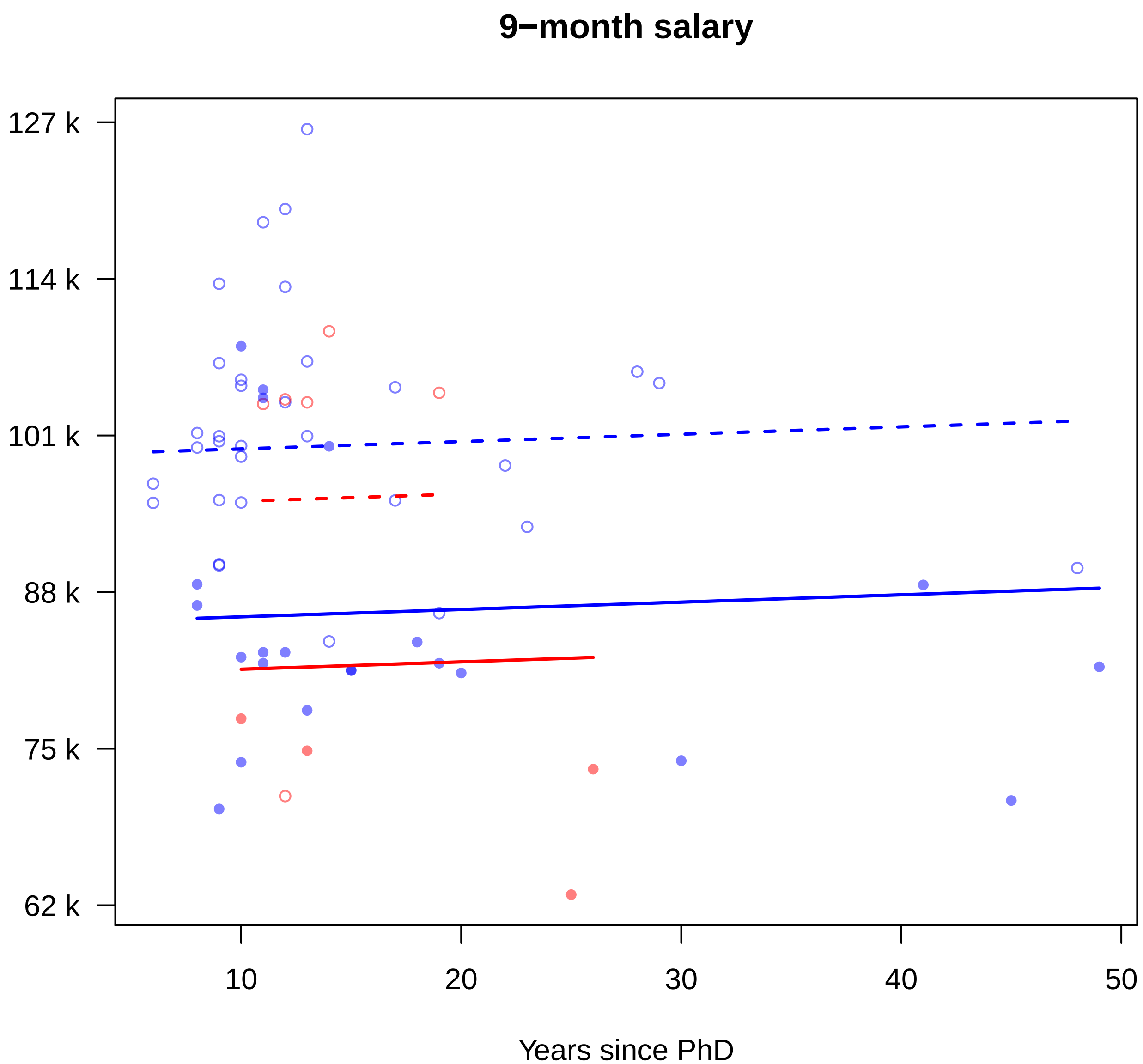
```r
attach(Salaries)
yr.range <- tapply(yrs.since.phd, list(discipline, sex, rank), range)
sex.col <- ifelse(sex == "Male", "blue", "red")
dis.col <- ifelse(discipline == "A", 16, 1)

beta0 <- m1$coefficients[1]
betaDisp <- m1$coefficients[2]
betaAssoc <- m1$coefficients[3]
betaProf <- m1$coefficients[4]
betaMale <- m1$coefficients[5]
beta1 <- m1$coefficients[6]

library(scales)
# Plot the model fits by rank
## Assist prof
assistant <- which(rank == "AsstProf")
plot(yrs.since.phd[assistant], salary[assistant], pch = dis.col[assistant], cex = 0.8,
     col = alpha(sex.col[assistant], 0.5), yaxt = "n", xlab = "Years since PhD",
     main = "9-month salary", ylab = "")
axis(2, at = seq(63000, 99000, len = 6), labels = paste(seq(63000, 99000, len = 6)/ 1000, "k"),
     las = 1)

segments(yr.range[[1]][1], beta0 + yr.range[[1]][1] * beta1,
         yr.range[[1]][2], beta0 + yr.range[[1]][2] * beta1, col = "red", lwd = 1.8)
segments(yr.range[[2]][1], beta0 + betaDisp + yr.range[[2]][1] * beta1,
```

```
         yr.range[[2]][2], beta0 + betaDisp + yr.range[[2]][2] * beta1,
         col = "red", lty = 2, lwd = 1.8)
segments(yr.range[[3]][1], beta0 + betaMale + yr.range[[3]][1] * beta1,
         yr.range[[3]][2], beta0 + betaMale + yr.range[[3]][2] * beta1,
         col = "blue", lwd = 1.8)
segments(yr.range[[4]][1], beta0 + betaDisp + betaMale + yr.range[[4]][1] * beta1,
         yr.range[[4]][2], beta0 + betaDisp + betaMale + yr.range[[4]][2] * beta1,
         col = "blue", lty = 2, lwd = 1.8)
```



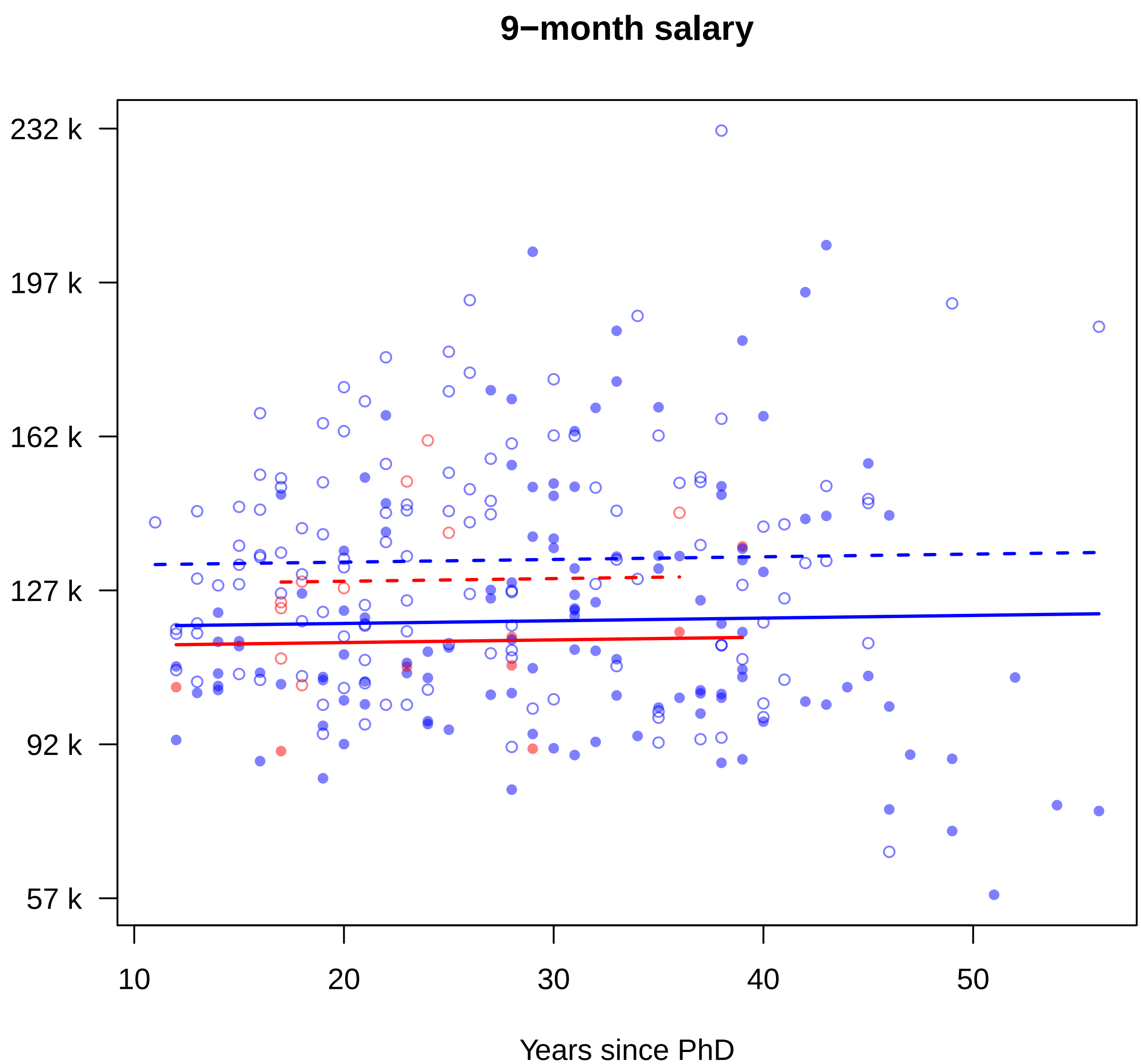

```
## Assoc prof
assoc <- which(rank == "AssocProf")
plot(yrs.since.phd[assoc], salary[assoc], pch = dis.col[assoc], cex = 0.8,
     col = alpha(sex.col[assoc], 0.5), yaxt = "n", xlab = "Years since PhD",
     main = "9-month salary", ylab = "")
axis(2, at = seq(62000, 127000, len = 6), labels = paste(seq(62000, 127000, len = 6)/ 1000, "k"),
     las = 1)
```

```r
segments(yr.range[[5]][1], beta0 + betaAssoc + yr.range[[5]][1] * beta1,
         yr.range[[5]][2], beta0 + betaAssoc + yr.range[[5]][2] * beta1,
         col = "red", lwd = 1.8)
segments(yr.range[[6]][1], beta0 + betaDisp + betaAssoc + yr.range[[6]][1] * beta1,
         yr.range[[6]][2], beta0 + betaDisp + betaAssoc + yr.range[[6]][2] * beta1,
         col = "red", lty = 2, lwd = 1.8)
segments(yr.range[[7]][1], beta0 + betaAssoc + betaMale + yr.range[[7]][1] * beta1,
         yr.range[[7]][2], beta0 + betaAssoc + betaMale + yr.range[[7]][2] * beta1,
         col = "blue", lwd = 1.8)
segments(yr.range[[8]][1], beta0 + betaDisp + betaAssoc + betaMale + yr.range[[8]][1] * beta1,
         yr.range[[8]][2], beta0 + betaDisp + betaAssoc + betaMale + yr.range[[8]][2] * beta1,
         col = "blue", lty = 2, lwd = 1.8)
```



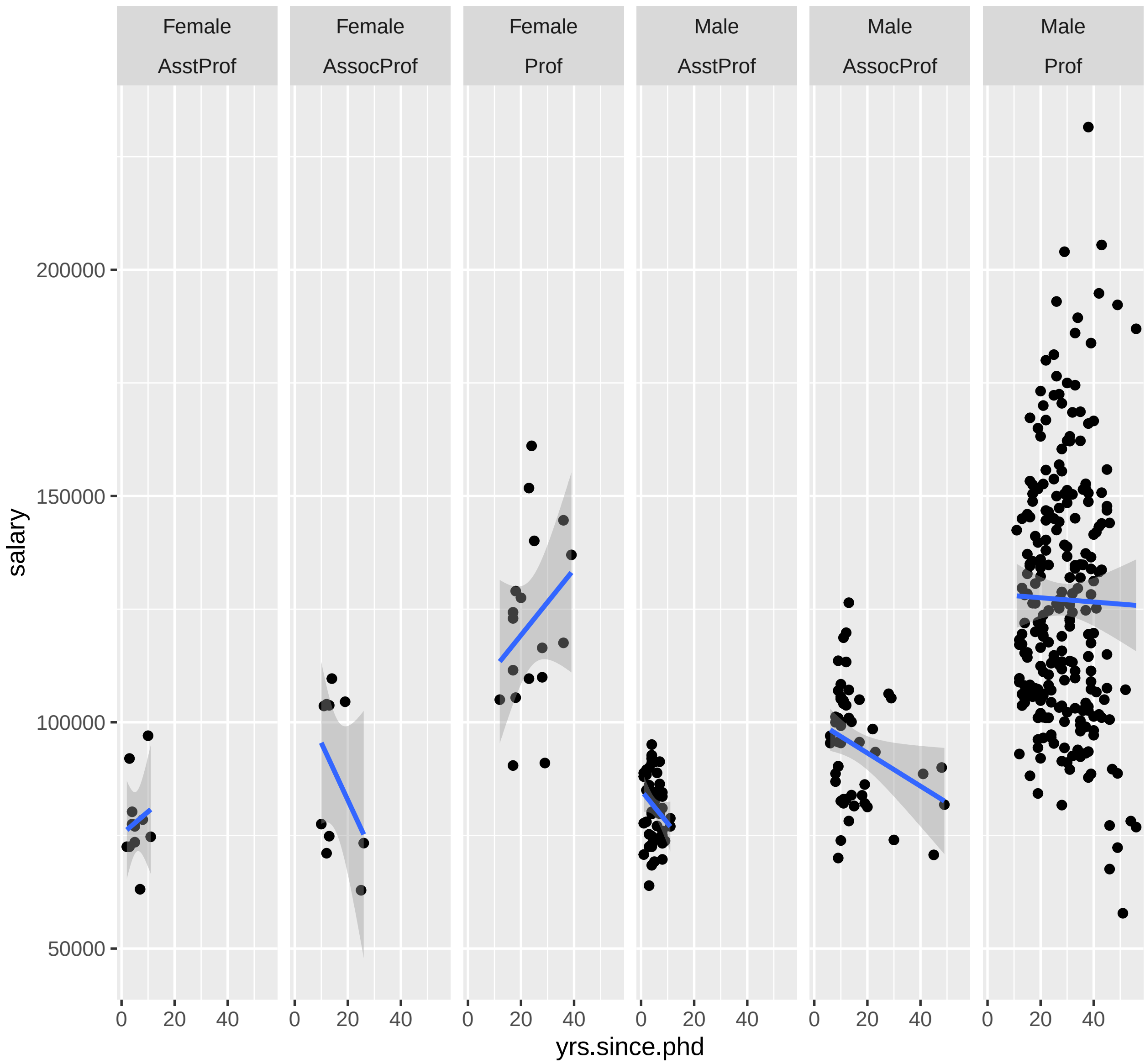

```r
## Full prof
prof <- which(rank == "Prof")
plot(yrs.since.phd[prof], salary[prof],
     pch = dis.col[prof], cex = 0.8,
```

```
     col = alpha(sex.col[prof], 0.5),
     yaxt = "n", xlab = "Years since PhD",
     main = "9-month salary", ylab = "")
axis(2, at = seq(57000, 232000, len = 6),
     labels = paste(seq(57000, 232000, len = 6)/ 1000, "k"),
     las = 1)
segments(yr.range[[9]][1], beta0 + betaProf + yr.range[[9]][1] * beta1,
         yr.range[[9]][2], beta0 + betaProf + yr.range[[9]][2] * beta1,
         col = "red", lwd = 1.8)
segments(yr.range[[10]][1], beta0 + betaDisp + betaProf + yr.range[[10]][1] * beta1,
         yr.range[[10]][2], beta0 + betaDisp + betaProf + yr.range[[10]][2] * beta1,
         col = "red", lty = 2, lwd = 1.8)
segments(yr.range[[11]][1], beta0 + betaProf + betaMale + yr.range[[11]][1] * beta1,
         yr.range[[11]][2], beta0 + betaProf + betaMale + yr.range[[11]][2] * beta1,
         col = "blue", lwd = 1.8)
segments(yr.range[[12]][1], beta0 + betaDisp + betaProf + betaMale + yr.range[[12]][1] * beta1,
         yr.range[[12]][2], beta0 + betaDisp + betaProf + betaMale + yr.range[[12]][2] * beta1,
         col = "blue", lty = 2, lwd = 1.8)
```




```
## Using ggplot
plot <- ggplot(aes(x = yrs.since.phd, y = salary), data = Salaries)
plot <- plot + geom_point()
plot <- plot + facet_grid(~ sex + rank)
(plot <- plot + geom_smooth(method = "lm"))
```

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



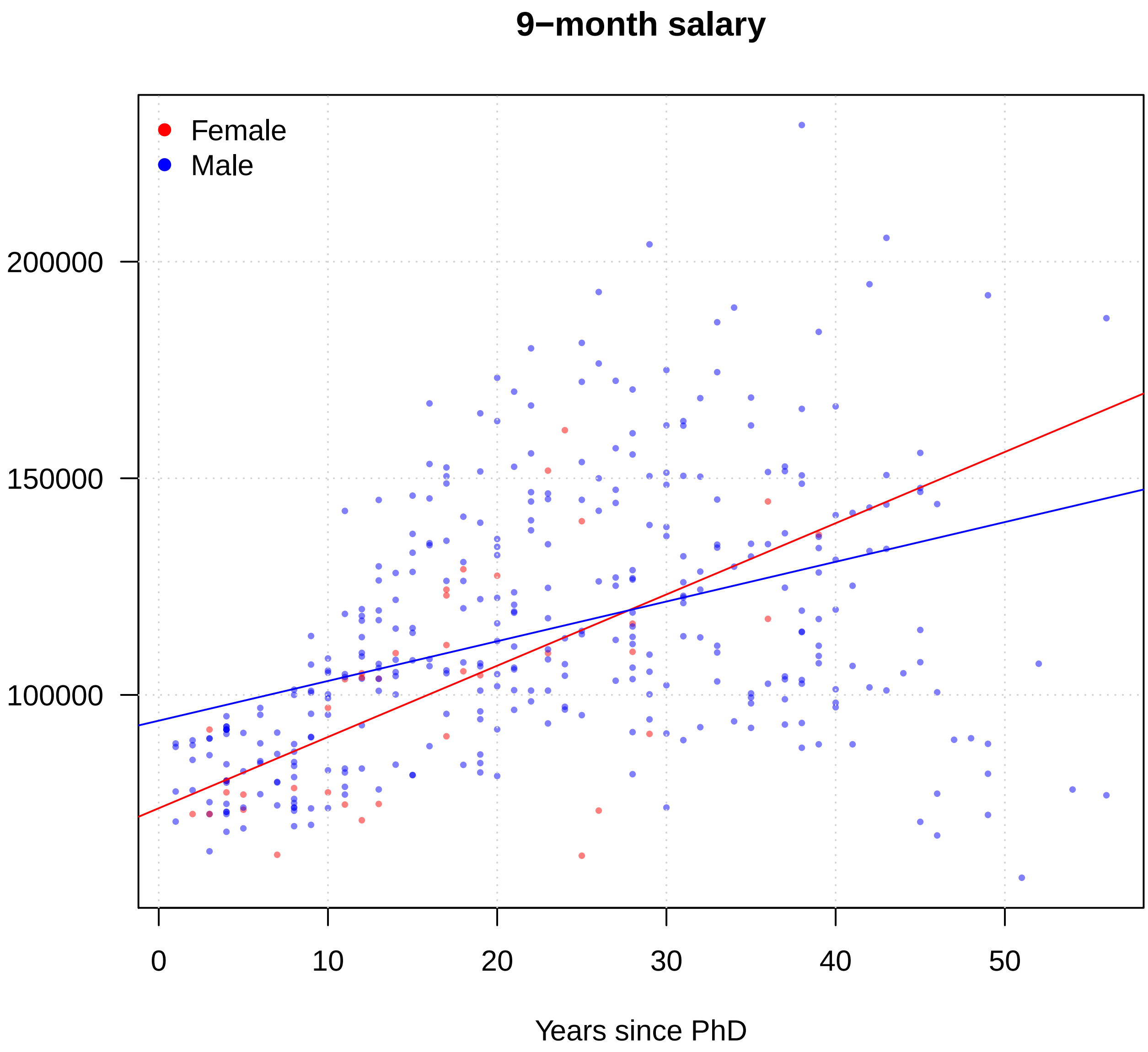
Model 2: Another MLR where we include the *interaction* between sex and `yrs.since.phd`

```
m2 <- lm(salary ~ sex * yrs.since.phd)
summary(m2)
```

```
##
## Call:
## lm(formula = salary ~ sex * yrs.since.phd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -83012 -19442  -2988  15059 102652
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              73840.8     8696.7   8.491 4.27e-16 ***
## sexMale                  20209.6     9179.2   2.202 0.028269 *
```

```
## yrs.since.phd           1644.9      454.6   3.618 0.000335 ***
## sexMale:yrs.since.phd   -728.0      468.0  -1.555 0.120665
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27420 on 393 degrees of freedom
## Multiple R-squared:  0.1867, Adjusted R-squared:  0.1805
## F-statistic: 30.07 on 3 and 393 DF,  p-value: < 2.2e-16
```

```
coeff <- m2$coefficients
plot(yrs.since.phd, salary, las = 1, pch = 16, cex = 0.5, col = alpha(sex.col, 0.5),
     xlab = "Years since PhD", main = "9-month salary", ylab = "")
grid()
abline(coeff[1], coeff[3], col = "red")
abline(coeff[1] + coeff[2], coeff[3] + coeff[4], col = "blue")
legend("topleft", legend = c("Female", "Male"),
       pch = 16, col = c("red", "blue"), bty = "n")
```

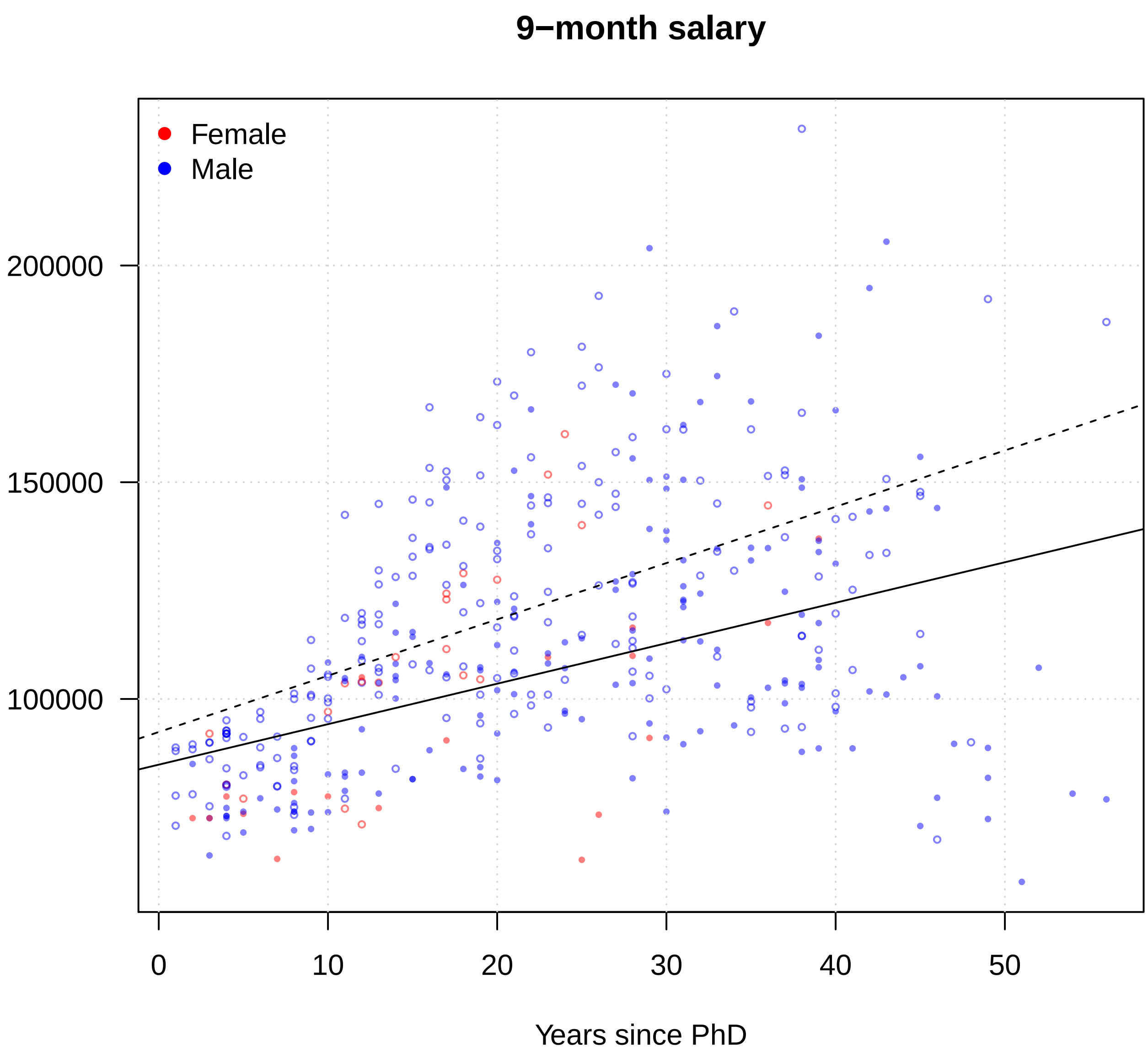


Model 3: One more MLR where we include the *interaction* between `discipline` and `yrs.since.phd`

```
m3 <- lm(salary ~ discipline * yrs.since.phd)
summary(m3)
```

```
##
## Call:
## lm(formula = salary ~ discipline * yrs.since.phd)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -84580 -16974  -3620  15733  92072
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               84845.4     4283.9  19.806  < 2e-16 ***
## disciplineB                7530.0     5492.2   1.371   0.1711
## yrs.since.phd               933.9      150.0   6.225 1.24e-09 ***
## disciplineB:yrs.since.phd   365.3      211.0   1.731   0.0842 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26400 on 393 degrees of freedom
## Multiple R-squared:  0.2458, Adjusted R-squared:  0.2401
## F-statistic:  42.7 on 3 and 393 DF,  p-value: < 2.2e-16
```

```
coeff <- m3$coefficients
plot(yrs.since.phd, salary, las = 1, pch = dis.col, cex = 0.5, col = alpha(sex.col, 0.5),
     xlab = "Years since PhD", main = "9-month salary", ylab = "")
grid()
abline(coeff[1], coeff[3])
abline(coeff[1] + coeff[2], coeff[3] + coeff[4], lty = 2)
legend("toplef", legend = c("Female", "Male"),
       pch = 16, col = c("red", "blue"), bty = "n")
```



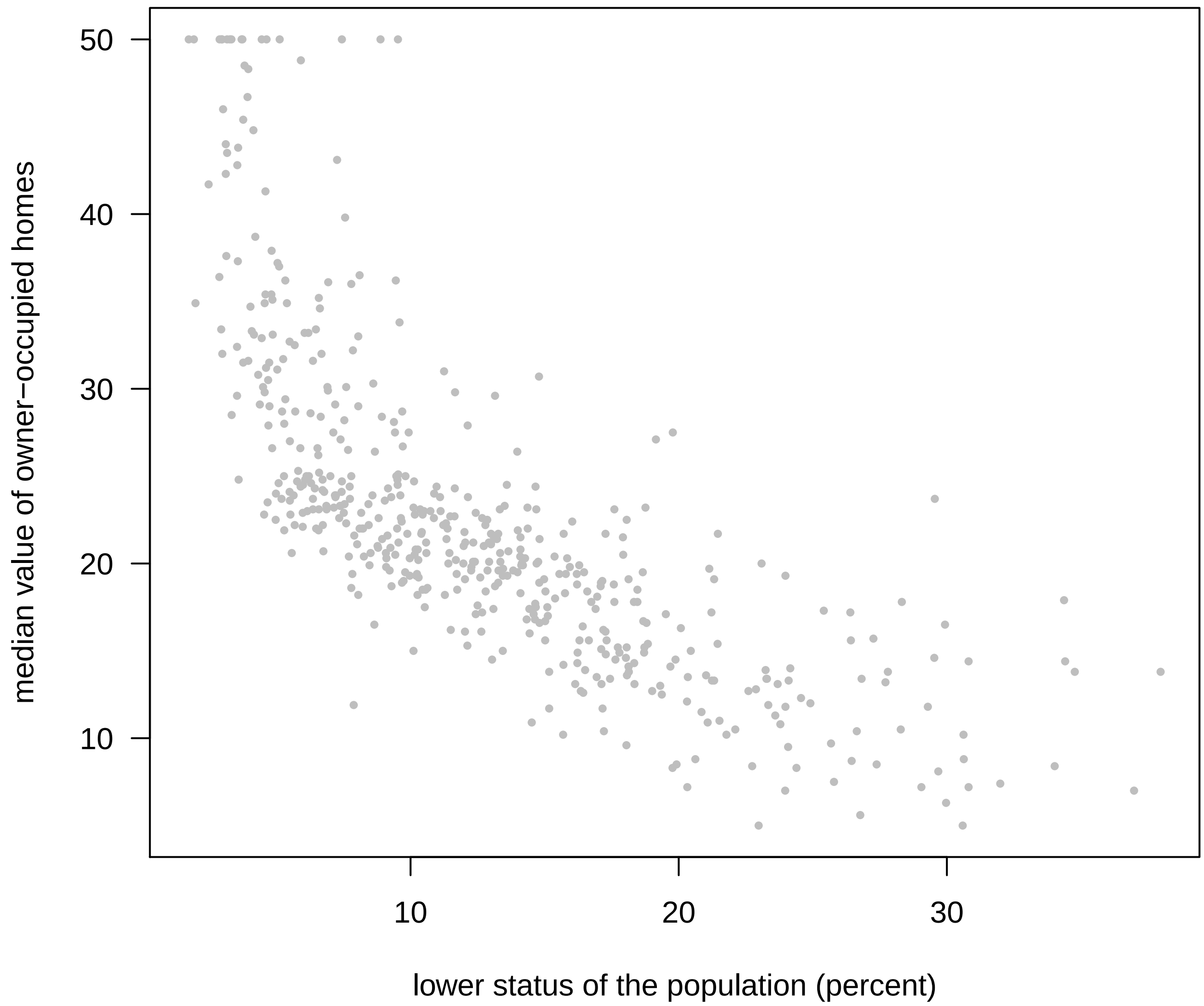
Polynomial regression

Housing Values in Suburbs of Boston

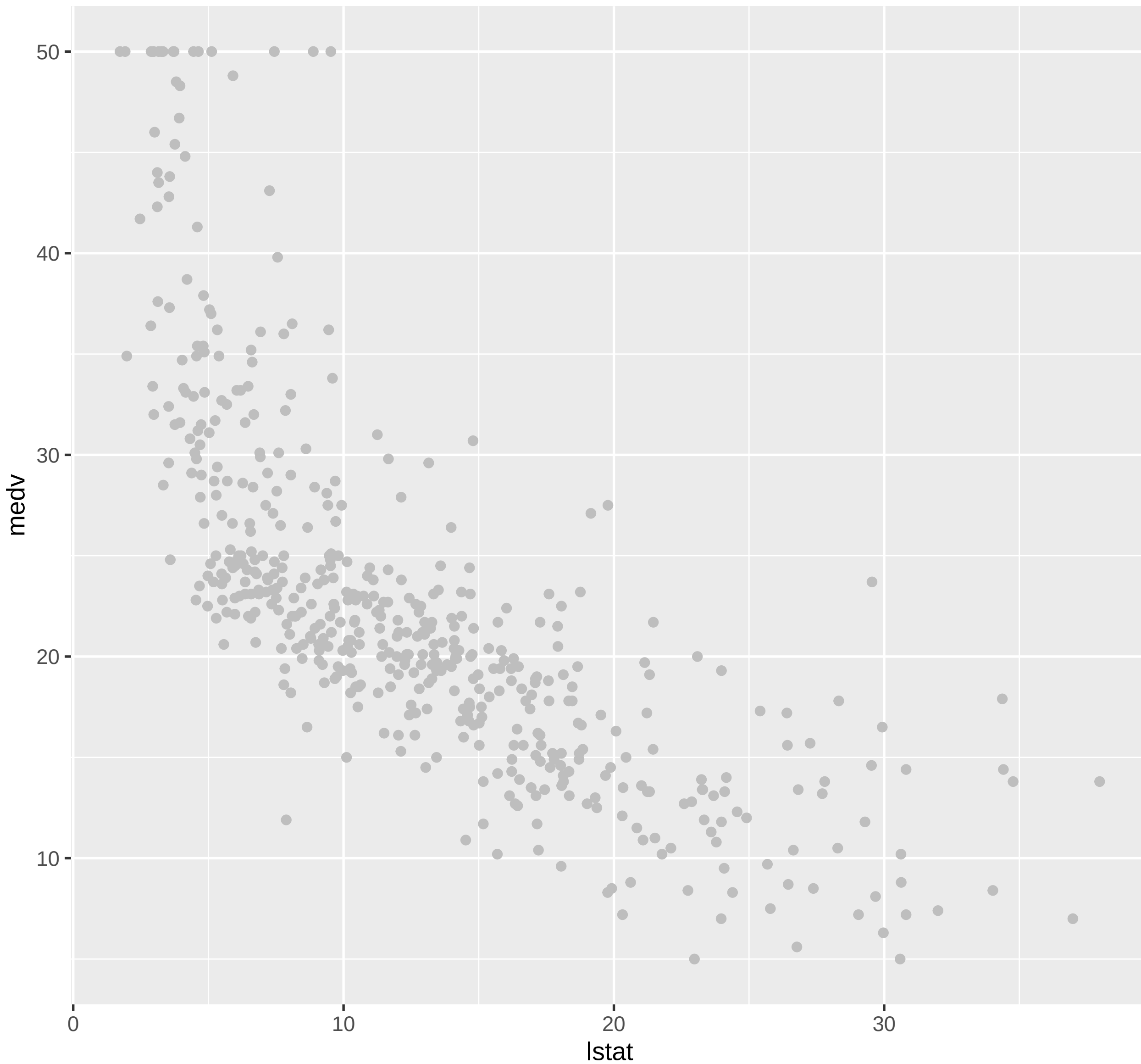
- Dependent variable: *medv*, the median value of owner-occupied homes (in thousands of dollars).
- Independent variable: *lstat* (percent of lower status of the population).

Load and plot the data

```
library(MASS)
data(Boston)
plot(Boston$lstat, Boston$medv, col = "gray", pch = 16,
     cex = 0.6, las = 1, xlab = "lower status of the population (percent)", ylab = "median value of owne
```

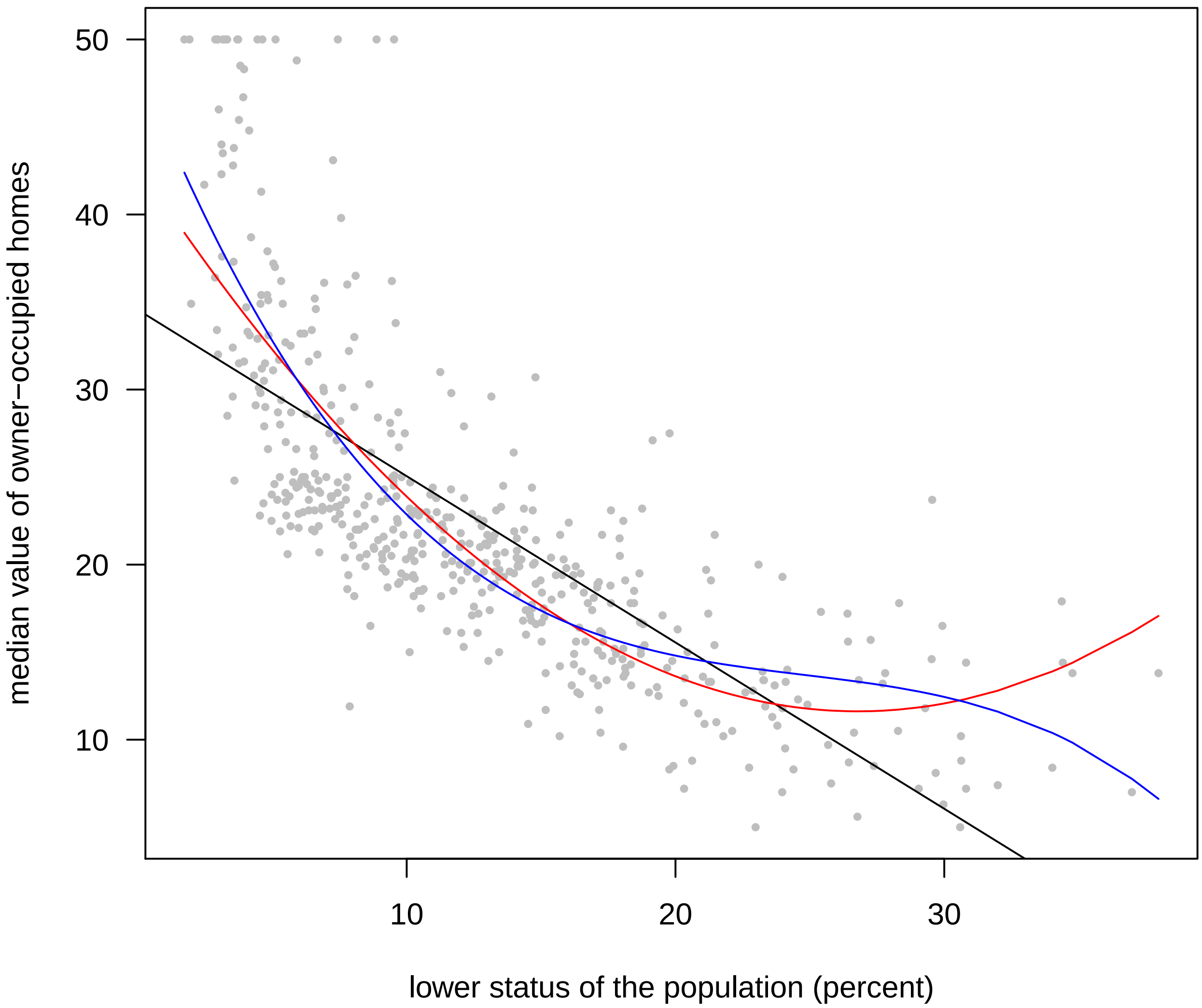



```
## ggplot
plot <- ggplot(aes(x = lstat, y = medv), data = Boston)
(plot <- plot + geom_point(colour = "gray"))
```



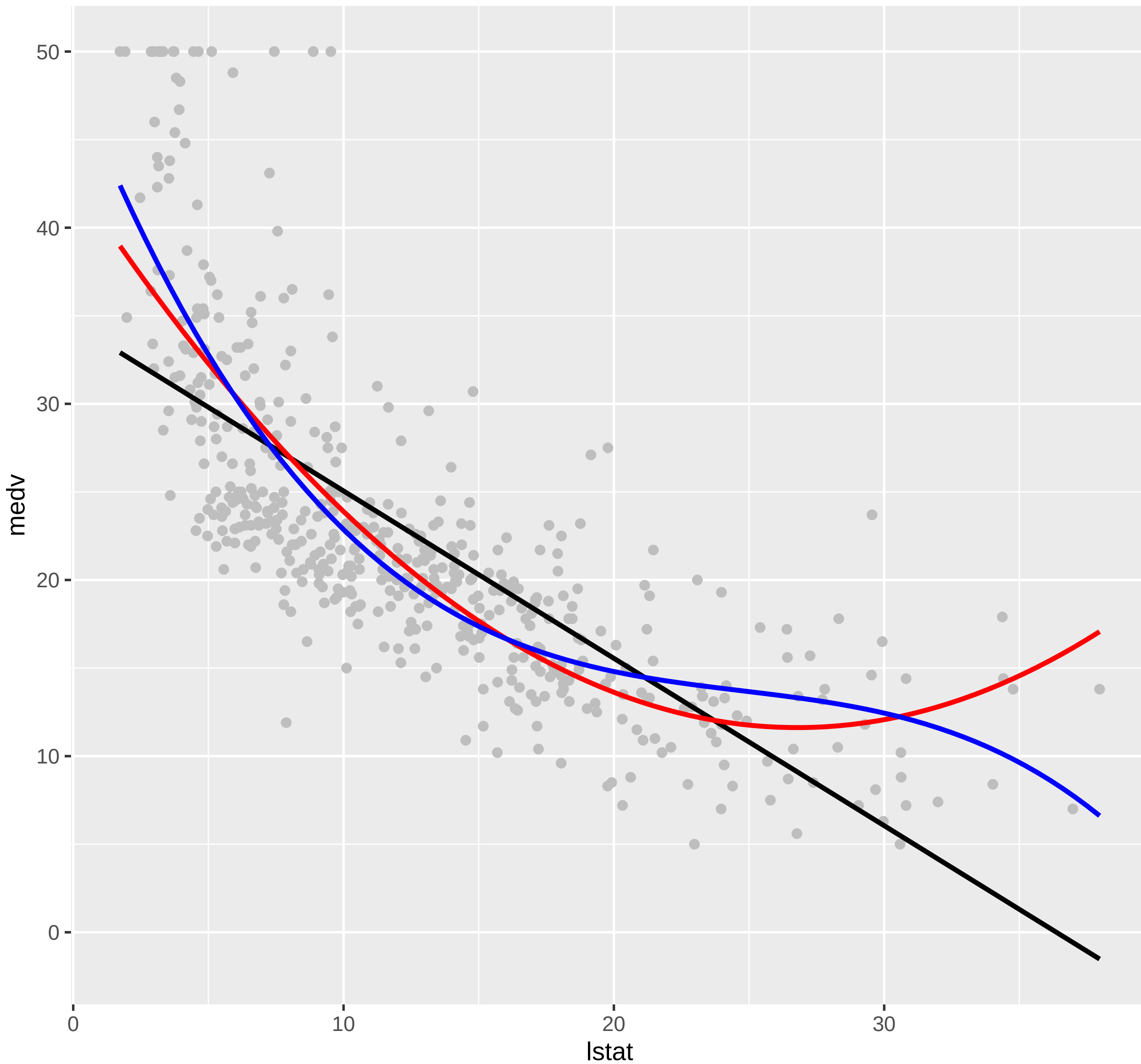
Plot the poylnomial regression fits

```
plot(Boston$lstat, Boston$medv, col = "gray", pch = 16,
     cex = 0.6, las = 1, xlab = "lower status of the population (percent)", ylab = "median value of owne
## SLR
m1 <- lm(medv ~ lstat, data = Boston)
abline(m1)
## 2nd order polynomial fit
m2 <- lm(medv ~ lstat + I(lstat^2), data = Boston)
lines(sort(Boston$lstat), m2$fitted.values[order(Boston$lstat)], col = "red")
## 3rd order polynomial fit
m3 <- lm(medv ~ lstat + I(lstat^2)+ I(lstat^3), data = Boston)
lines(sort(Boston$lstat), m3$fitted.values[order(Boston$lstat)], col = "blue")
```




```r
## Using ggplot
plot <- plot + geom_smooth(method = "lm", colour = "black", se = F)
plot <- plot + geom_smooth(method = "lm", formula = y ~ x + I(x^2), colour = "red", se = F)
plot <- plot + geom_smooth(method = "lm", formula = y ~ x + I(x^2) + I(x^3), colour = "blue", se = F)
plot
```

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



Model selection

```r
anova(m2, m3)
```

```
## Analysis of Variance Table
##
## Model 1: medv ~ lstat + I(lstat^2)
## Model 2: medv ~ lstat + I(lstat^2) + I(lstat^3)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    503 15347
## 2    502 14616  1    731.76 25.134 7.428e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## Use Orthogonal Polynomials
m2new <- lm(medv ~ poly(lstat, 2), data = Boston)
```

```r
m3new <- lm(medv ~ poly(lstat, 3), data = Boston)
summary(m3new); summary(m3)
```

```
##
## Call:
## lm(formula = medv ~ poly(lstat, 3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5441  -3.7122  -0.5145   2.4846  26.4153
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.5328     0.2399  93.937  < 2e-16 ***
## poly(lstat, 3)1 -152.4595     5.3958 -28.255  < 2e-16 ***
## poly(lstat, 3)2   64.2272     5.3958  11.903  < 2e-16 ***
## poly(lstat, 3)3  -27.0511     5.3958  -5.013 7.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.396 on 502 degrees of freedom
## Multiple R-squared:  0.6578, Adjusted R-squared:  0.6558
## F-statistic: 321.7 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2) + I(lstat^3), data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5441  -3.7122  -0.5145   2.4846  26.4153
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 48.6496253  1.4347240  33.909  < 2e-16 ***
## lstat       -3.8655928  0.3287861 -11.757  < 2e-16 ***
## I(lstat^2)   0.1487385  0.0212987   6.983 9.18e-12 ***
## I(lstat^3)  -0.0020039  0.0003997  -5.013 7.43e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.396 on 502 degrees of freedom
## Multiple R-squared:  0.6578, Adjusted R-squared:  0.6558
## F-statistic: 321.7 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
anova(m2new, m3new)
```

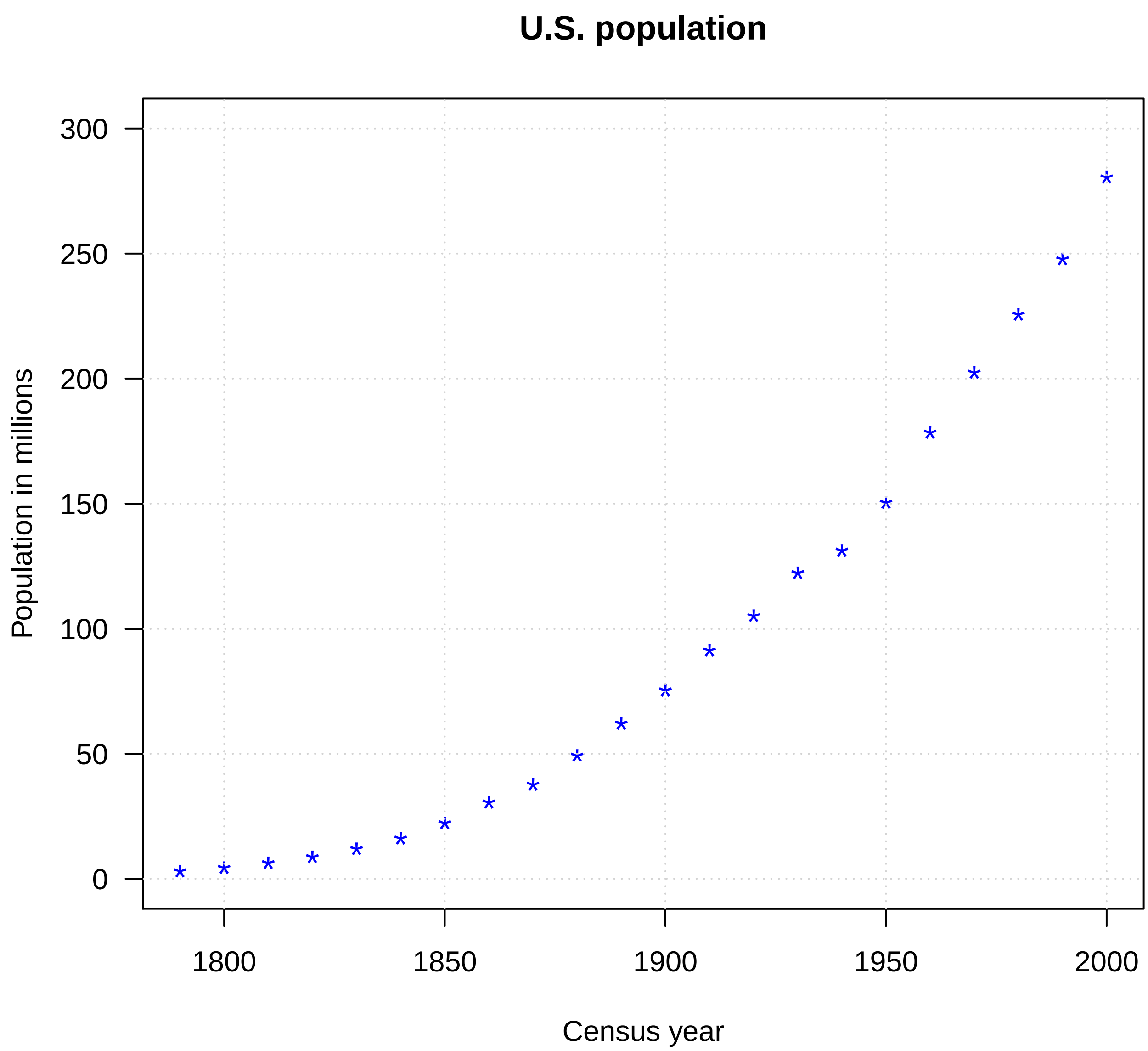
```
## Analysis of Variance Table
##
## Model 1: medv ~ poly(lstat, 2)
## Model 2: medv ~ poly(lstat, 3)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    503 15347
## 2    502 14616  1    731.76 25.134 7.428e-07 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nonlinear Regression

U.S. Population Example

```
library(car)
plot(population ~ year, data = USPop, main = "U.S. population",
     ylim = c(0, 300),pch = "*", xlab = "Census year",
     ylab = "Population in millions", cex = 1.25, las = 1, col = "blue")
grid()
```



Logistic growth curve

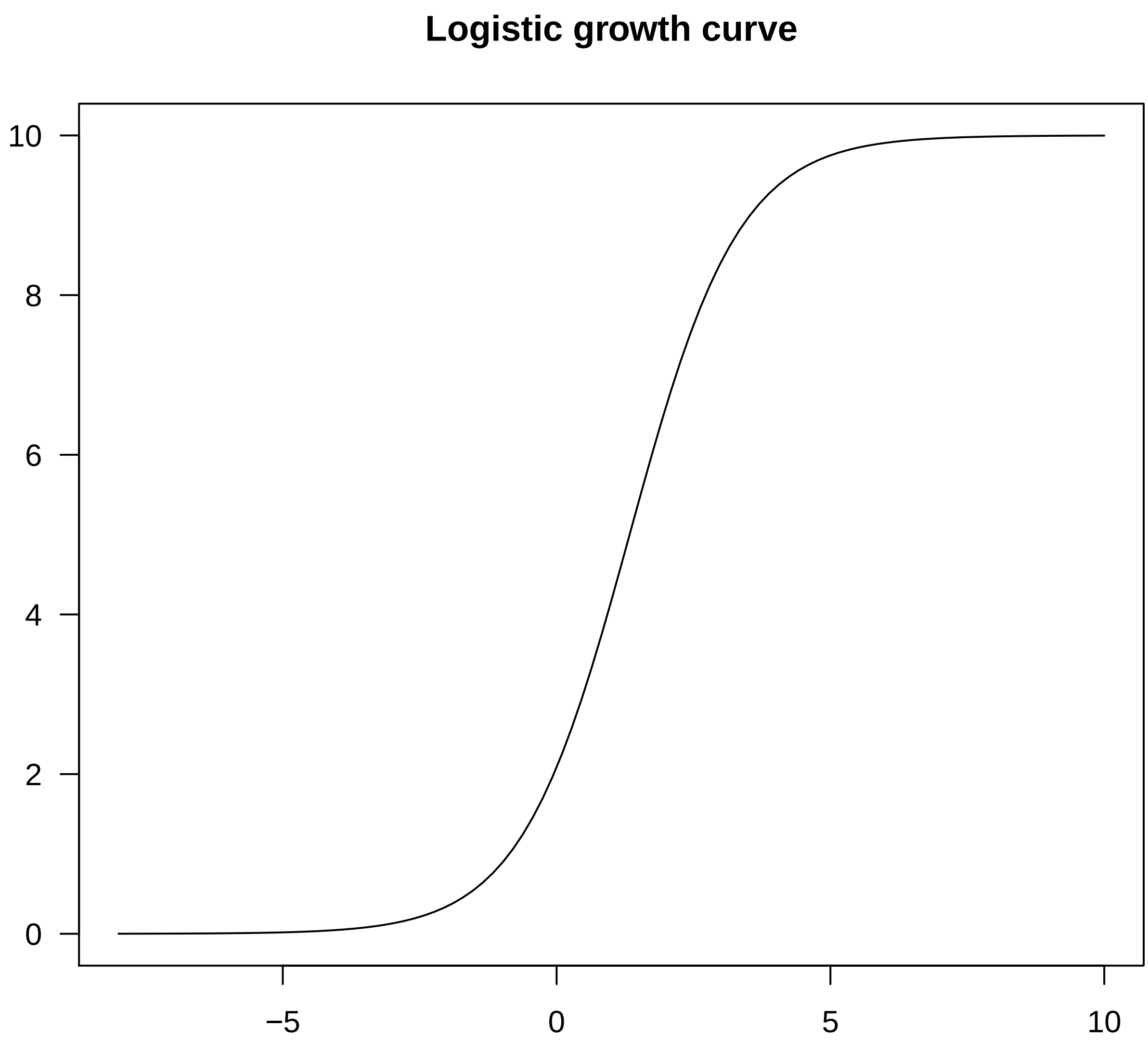
A logistic function is a symmetric S shape curve with equation:

$$f(x) = \frac{\phi_1}{1 + \exp(-(x - \phi_2)/\phi_3)}$$

where ϕ_1 is the curve's maximum value; ϕ_2 is the curve's midpoint in x; and ϕ_3 is the "range" (or the inverse growth rate) of the curve.

One typical application of the logistic equation is to model population growth.

```r
# phi_1 = 10; phi_2 = 4/3, phi_3 = 1
curve(10 / (1 + exp(-(x - 4/3))), from = -8, to = 10, main = "Logistic growth curve",
      las = 1, xlab = "", ylab = "")
```

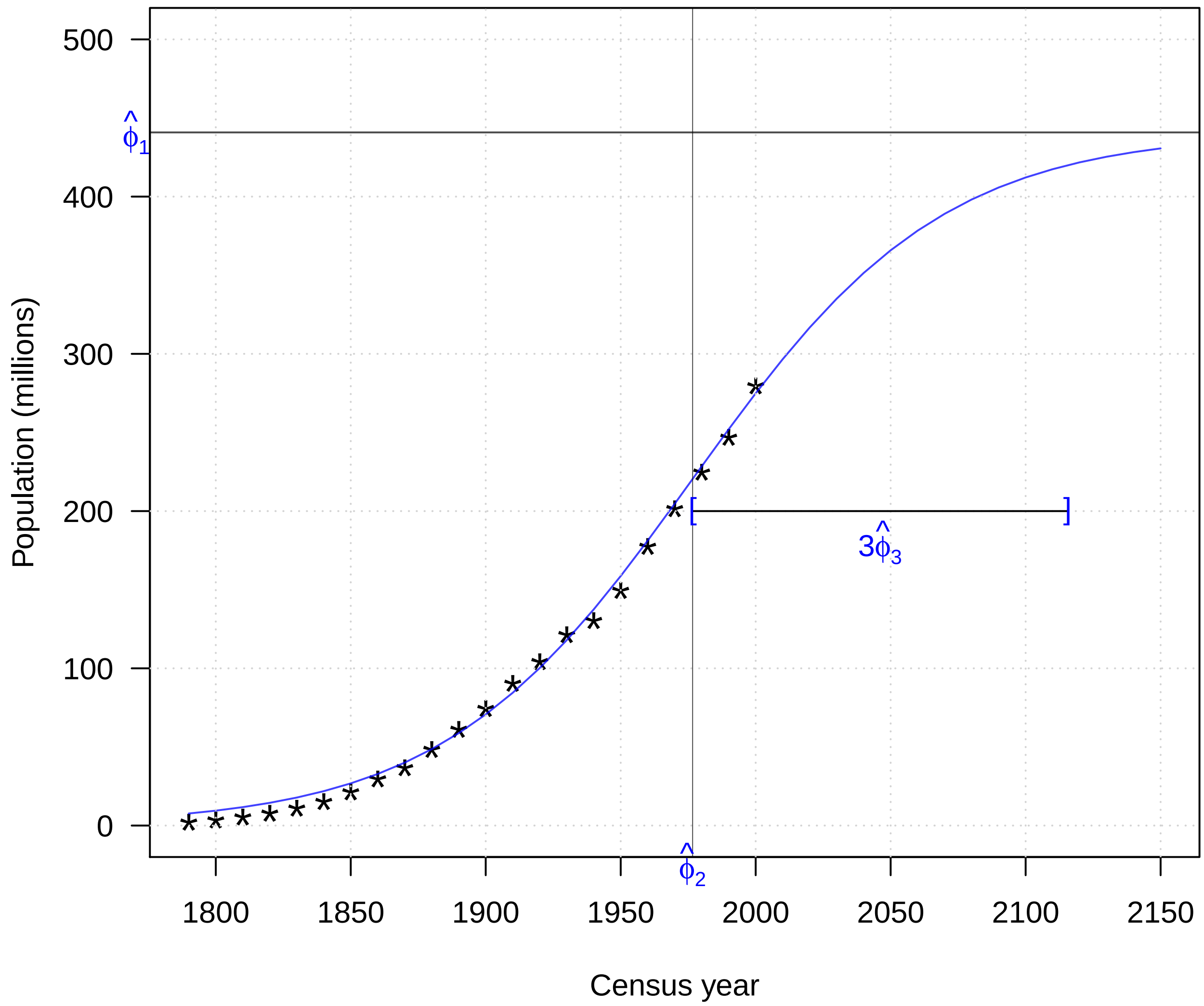


Fit a logistic growth curve to the U.S. population data set

```r
pop.ss <- nls(population ~ SSlogis(year, phi1, phi2, phi3), data = USPop)
summary(pop.ss)
```

```
##
## Formula: population ~ SSlogis(year, phi1, phi2, phi3)
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## phi1  440.833     35.000   12.60 1.14e-10 ***
## phi2 1976.634      7.556  261.61  < 2e-16 ***
## phi3   46.284      2.157   21.45 8.87e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.909 on 19 degrees of freedom
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 6.818e-07
```

```r
library(scales)

plot(population ~ year, USPop, xlim = c(1790, 2150),
     ylim = c(0, 500), las = 1, pch = "*",
     xlab = "Census year", ylab = "Population (millions)", cex = 1.6)
with(USPop, lines(seq(1790, 2150, by = 10),
                  predict(pop.ss, data.frame(year = seq(1790, 2150, by = 10))),
                  lwd = 1, col = alpha("blue", 0.75)))
abline(h = coef(pop.ss)[1], col = alpha("black", 0.7))
mtext(expression(hat(phi)[1]), side = 2, at = coef(pop.ss)[1], las = 1, col = "blue")
grid()
abline(v = coef(pop.ss)[2], col = alpha("black", 0.7), lwd = 0.5)
mtext(expression(hat(phi)[2]), side = 1, at = coef(pop.ss)[2], las = 1, col = "blue")
segments(coef(pop.ss)[2], 200, coef(pop.ss)[2] + 3 * coef(pop.ss)[3])
text(coef(pop.ss)[2], 200, "[", col = "blue")
text(coef(pop.ss)[2] + 3 * coef(pop.ss)[3], 200, "]", col = "blue")
text(coef(pop.ss)[2] + 1.5 * coef(pop.ss)[3], 180, expression(3*hat(phi)[3]), col = "blue")
```




```r
# Compute AIC
AIC(pop.ss)
```

```
## [1] 137.2121
```

Alternative model: fit quadratic/cubic polynomial regression

```r
pop.qm <- lm(population ~ year + I(year^2), USPop)
pop.cm <- lm(population ~ poly(year, 3), USPop)
summary(pop.cm)
```

```
##
## Call:
## lm(formula = population ~ poly(year, 3), data = USPop)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2647 -1.1481  0.4461  1.7754  4.1953
##
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      94.6753     0.6023  157.20   <2e-16 ***
## poly(year, 3)1 383.5304     2.8249  135.77   <2e-16 ***
## poly(year, 3)2 112.4650     2.8249   39.81   <2e-16 ***
## poly(year, 3)3   5.1987     2.8249    1.84   0.0823 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.825 on 18 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.999
## F-statistic:  6674 on 3 and 18 DF,  p-value: < 2.2e-16
```

```
## Model selection
AIC(pop.cm); AIC(pop.qm)
```

```
## [1] 113.711
```

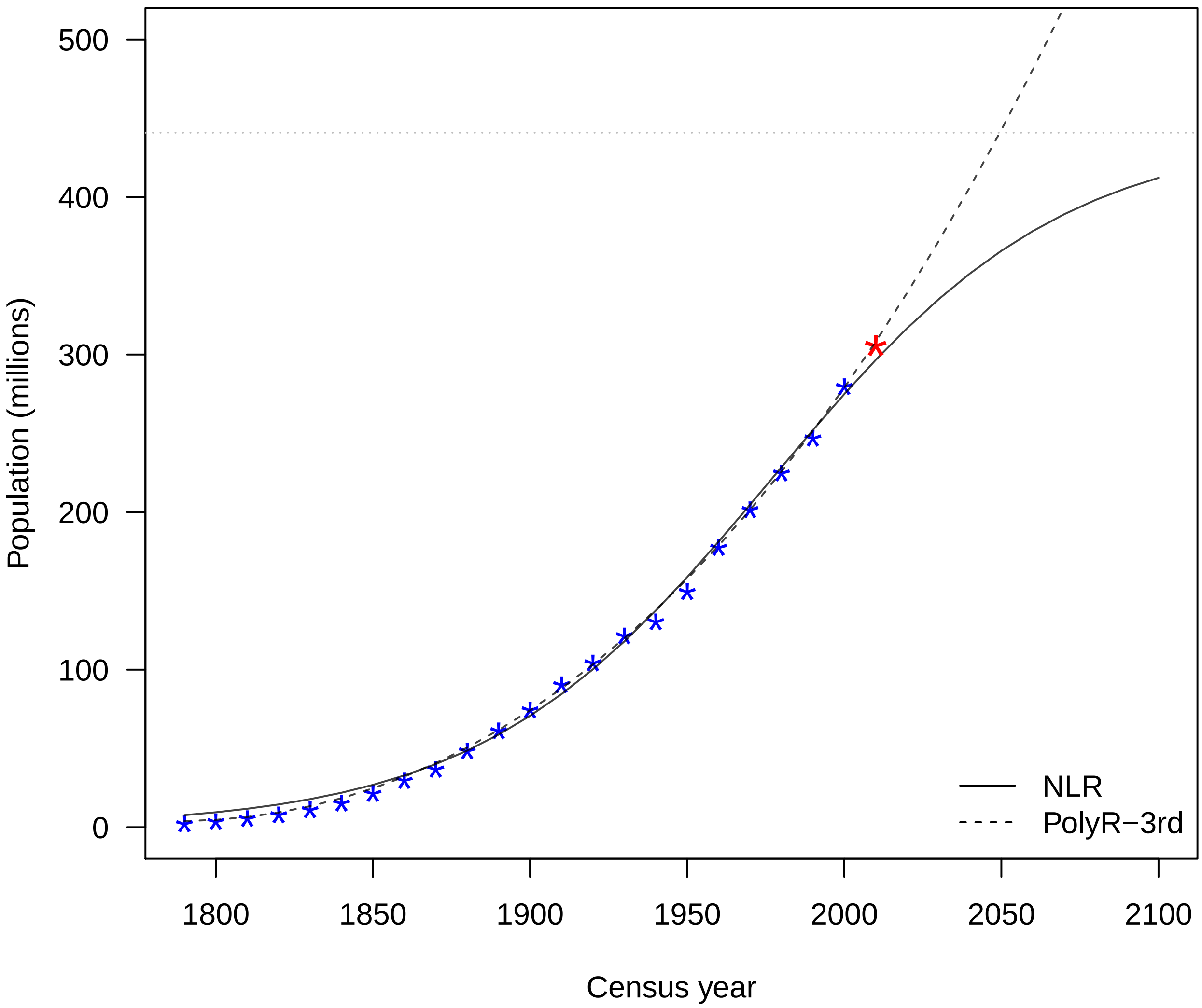
```
## [1] 115.5039
```

```
anova(pop.qm, pop.cm)
```

```
## Analysis of Variance Table
##
## Model 1: population ~ year + I(year^2)
## Model 2: population ~ poly(year, 3)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     19 170.66
## 2     18 143.64  1    27.027 3.3868 0.08227 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing the fits

```
plot(population ~ year, USPop, xlim = c(1790, 2100),
     ylim = c(0, 500), las = 1, pch = "*", col = "blue",
     xlab = "Census year", ylab = "Population (millions)", cex = 1.6)
with(USPop, lines(seq(1790, 2100, by = 10),
                  predict(pop.ss, data.frame(year = seq(1790, 2100, by = 10))),
                  lwd = 1, col = alpha("black", 0.75)))
points(2010, 308, pch = "*", cex = 2, col = "red")
abline(h = coef(pop.ss)[1], lty = 3, col = "gray", lwd = 0.95)
with(USPop, lines(seq(1790, 2100, by = 10), predict(pop.cm, data.frame(year = seq(1790, 2100, by = 10)))
                  lwd = 1, lty = 2, col = alpha("black", 0.75)))
legend("bottomright", legend = c("NLR", "PolyR-3rd"), lty = c(1, 2), bty = "n")
```



Population (millions)
500
400
300
200
100
0
1800
1850
1900
1950
2000
2050
2100
Census year
NLR
PolyR−3rd