Aufgabe 1

```
set.seed(123)
gendata = function (x, beta, len) {
  return (beta * log(x) + rnorm(len))
}

alllin <- matrix(NA,nrow=10000,ncol=7)
colnames(alllin) <- c("fdach","y", "quadVor","EBias", "VarFDach","EQuadVor", "AIC")
allpol <- matrix(NA,nrow=10000,ncol=7)
colnames(allpol) <- c("fdach","y", "quadVor","EBias", "VarFDach","EQuadVor", "AIC")
for (i in 1:10000){
  x <- runif(100,1,5)
  residuen <- rnorm(100,0,1)
  b <- 4
  y <- b*log(x)+residuen
  lmlinear <- lm(y~x)
  lmpoly <- lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5))
  alllin[i,1] <- lmlinear$coefficients  %*% c(1,3)
  allpol[i,1] <- lmpoly$coefficients %*% c(1,3,3^2,3^3,3^4,3^5)

  alllin[i,2] <- b*log(3)+rnorm(1)
  allpol[i,2] <- alllin[i,2]

  alllin[i,3] <- (alllin[i,2]-alllin[i,1])^2
  allpol[i,3] <- (allpol[i,2]-allpol[i,1])^2
  alllin[i,7] <- AIC(object = lmlinear)
  allpol[i,7] <- AIC(object = lmpoly)
}
alllin[,4] <- mean(alllin[,1])-4*log(3)
allpol[,4] <- mean(allpol[,1])-4*log(3)

alllin[,5] <- var(alllin[,1])
allpol[,5] <- var(allpol[,1])


x3 <- rep(3,10000)
y3 <- gendata(x3, 4, 10000)
alllin[,6] <- mean((y3-alllin[,1])^2)
allpol[,6] <- mean((y3-allpol[,1])^2)

AIC100_lin <- mean(alllin[,7])
AIC100_poly <- mean(allpol[,7])
```

```
alllin20 <- matrix(NA,nrow=10000,ncol=7)
colnames(alllin20) <- c("fdach","y", "quadVor","EBias", "VarFDach","EQuadVor", "AIC")
allpol20 <- matrix(NA,nrow=10000,ncol=7)
colnames(allpol20) <- c("fdach","y", "quadVor","EBias", "VarFDach","EQuadVor", "AIC")
for (i in 1:10000){
  x <- runif(20,1,5)
  residuen <- rnorm(20,0,1)
  b <- 4
  y <- b*log(x)+residuen
```

```
  lmlinear <- lm(y~x)
  lmpoly <- lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5))
  alllin20[i,1] <- lmlinear$coefficients  %*% c(1,3)
  allpol20[i,1] <- lmpoly$coefficients %*% c(1,3,3^2,3^3,3^4,3^5)

  alllin20[i,2] <- b*log(3)+rnorm(1)
  allpol20[i,2] <- alllin20[i,2]

  alllin20[i,3] <- (alllin20[i,2]-alllin20[i,1])^2
  allpol20[i,3] <- (allpol20[i,2]-allpol20[i,1])^2

  alllin20[i,7] <- AIC(object = lmlinear)
  allpol20[i,7] <- AIC(object = lmpoly)
}

alllin20[,4] <- mean(alllin20[,1])-4*log(3)
allpol20[,4] <- mean(allpol20[,1])-4*log(3)

alllin20[,5] <- var(alllin20[,1])
allpol20[,5] <- var(allpol20[,1])

x4 <- rep(3,20)
y4 <- gendata(x4, 4, 20)
alllin20[,6] <- mean((y4-alllin20[,1])^2)
allpol20[,6] <- mean((y4-allpol20[,1])^2)

AIC20_lin <- mean(alllin20[,7])
AIC20_poly <- mean(allpol20[,7])
```

```
cat("AIC (Linear):")
#> AIC (Linear):
cat("\n")
cat("AIC (n=100)")
#> AIC (n=100)
cat("\n")
AIC100_lin
#> [1] 297.6641
cat("\n")
cat("AIC (n=20)")
#> AIC (n=20)
cat("\n")
AIC20_lin
#> [1] 61.74052
cat("________________________")
#> ________________________
cat("\n")
cat("AIC (Polynomial):")
#> AIC (Polynomial):
cat("\n")
cat("AIC (n=100)")
#> AIC (n=100)
cat("\n")
AIC100_poly
#> [1] 290.4355
```

```
cat("\n")
cat("AIC (n=20)")
#> AIC (n=20)
cat("\n")
AIC20_poly
#> [1] 62.19022
```

```
cat("Quadratischer Vorhersagefehler (Linear):")
#> Quadratischer Vorhersagefehler (Linear):
cat("\n")
cat("Quadratischer Vorhersagefehler (n=100)")
#> Quadratischer Vorhersagefehler (n=100)
cat("\n")
head(alllin[,6],1)
#> [1] 1.139113
cat("\n")
cat("Quadratischer Vorhersagefehler (n=20)")
#> Quadratischer Vorhersagefehler (n=20)
cat("\n")
head(alllin20[,6],1)
#> [1] 0.9740106
cat("________________________")
#> ________________________
cat("\n")
cat("Quadratischer Vorhersagefehler (Polynomial):")
#> Quadratischer Vorhersagefehler (Polynomial):
cat("\n")
cat("Quadratischer Vorhersagefehler (n=100)")
#> Quadratischer Vorhersagefehler (n=100)
cat("\n")
head(allpol[,6],1)
#> [1] 1.049011
cat("\n")
cat("Quadratischer Vorhersagefehler (n=20)")
#> Quadratischer Vorhersagefehler (n=20)
cat("\n")
head(allpol20[,6],1)
#> [1] 0.9853232
```

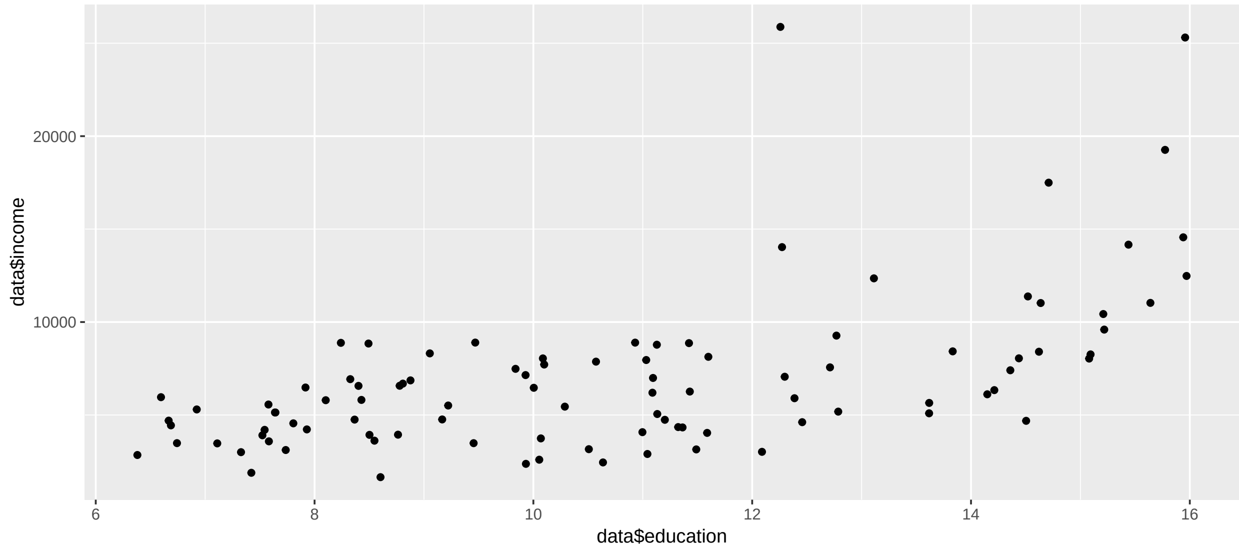
Der durchschnittliche AIC im Falle n=20 ist für das lineare Modell mit $AIC = 61.68768$ kleiner als im polynomialen Modell mit $AIC = 62.13233$. Im Falle n=100 ist das polynomiale Modell laut dem AIC besser ($AIC = 290.6327$) als das lineare Modell ($AIC = 297.949$). Dies stimmt mit den Ergebnissen der geschätzten quadratischen Vorhersagefehlern überein. Bei n=20 ist das lineare Modell leicht besser, im Falle n=100 produziert das polynomiale Modell die genaueren Vorhersagen.

Aufgabe 2

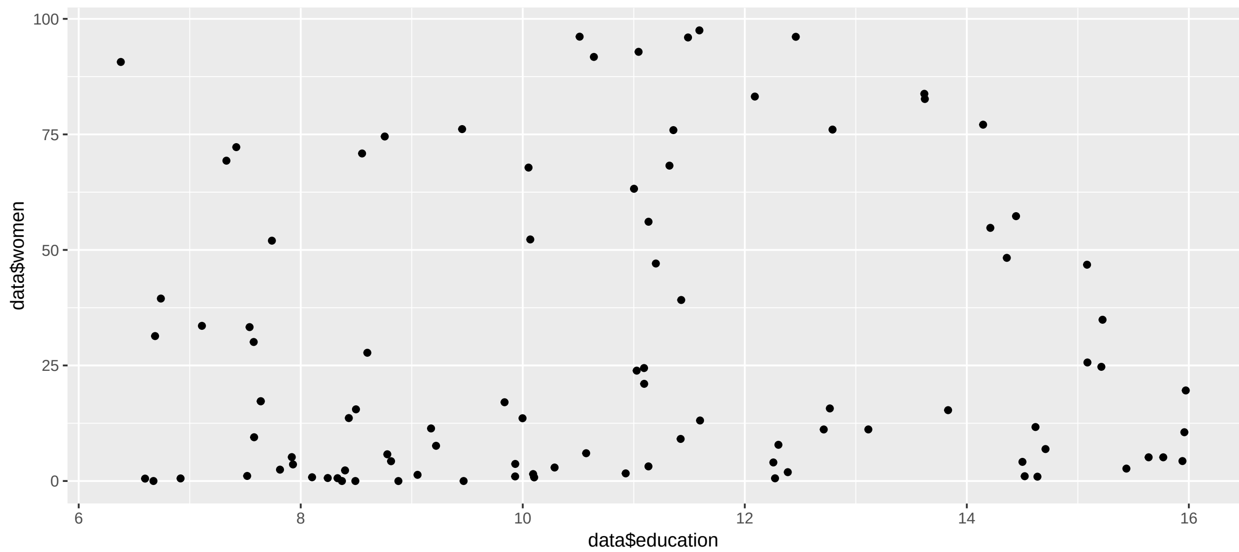
a)

```
library(car)
library(ggplot2)
data <- Prestige
```

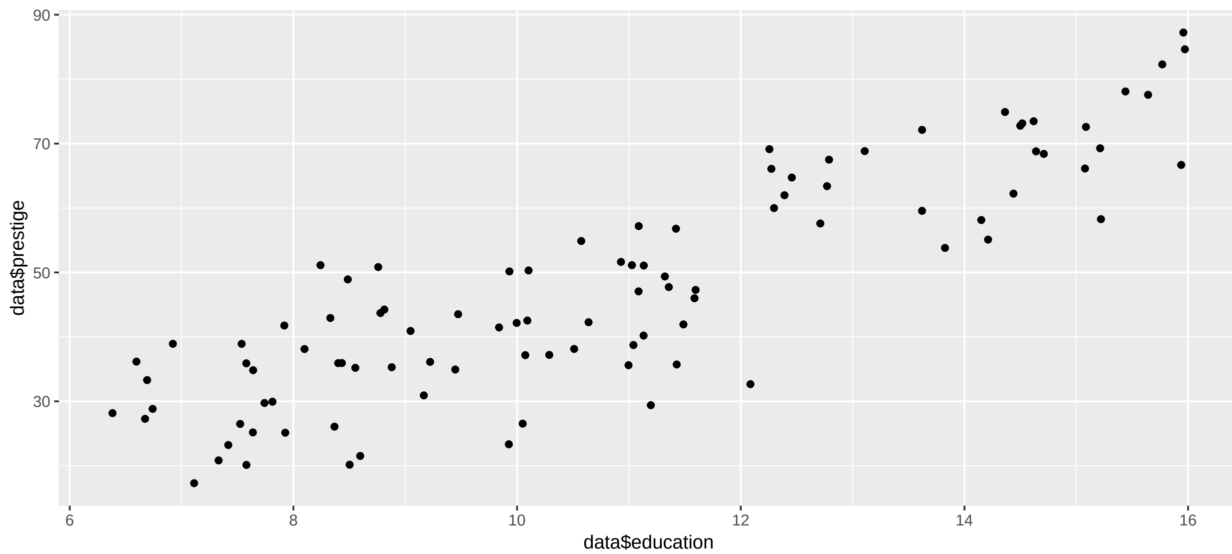
```
data$census <- NULL
data <- data[complete.cases(data),]
data$type1 <- as.numeric((data$type))
ggplot(data=data,aes(x=data$education,y=data$income))+geom_jitter()
```



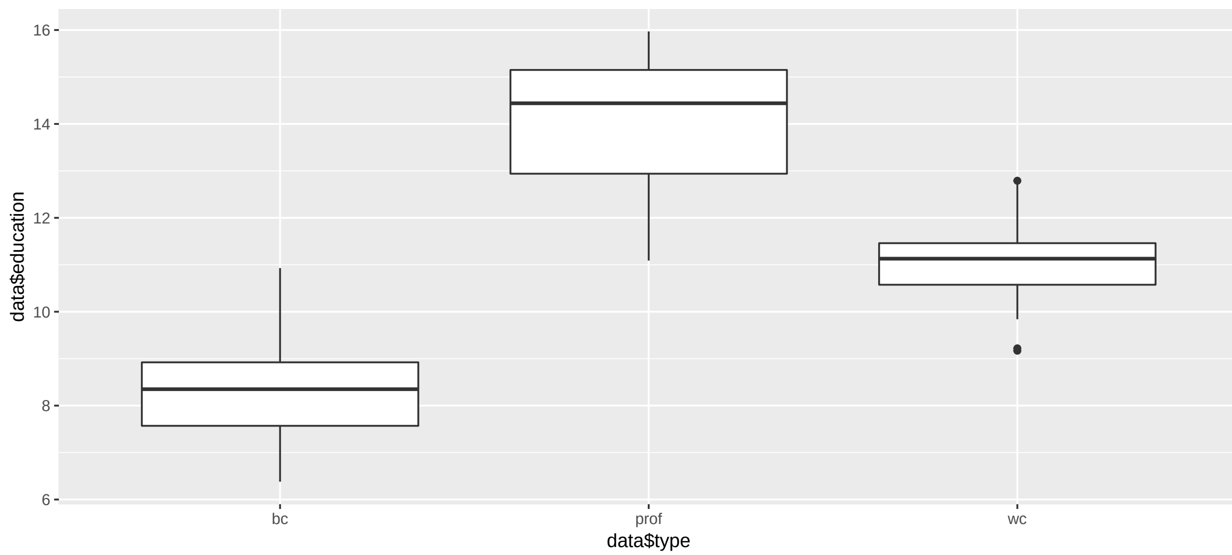

```
ggplot(data=data,aes(x=data$education,y=data$women))+geom_jitter()
```



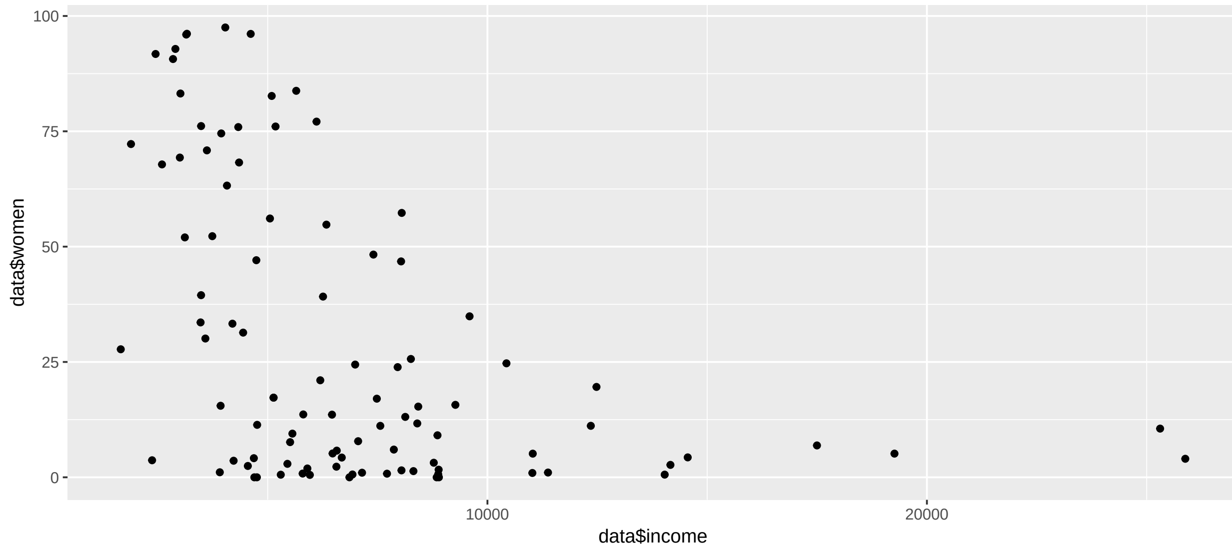

```
ggplot(data=data,aes(x=data$education,y=data$prestige))+geom_jitter()
```



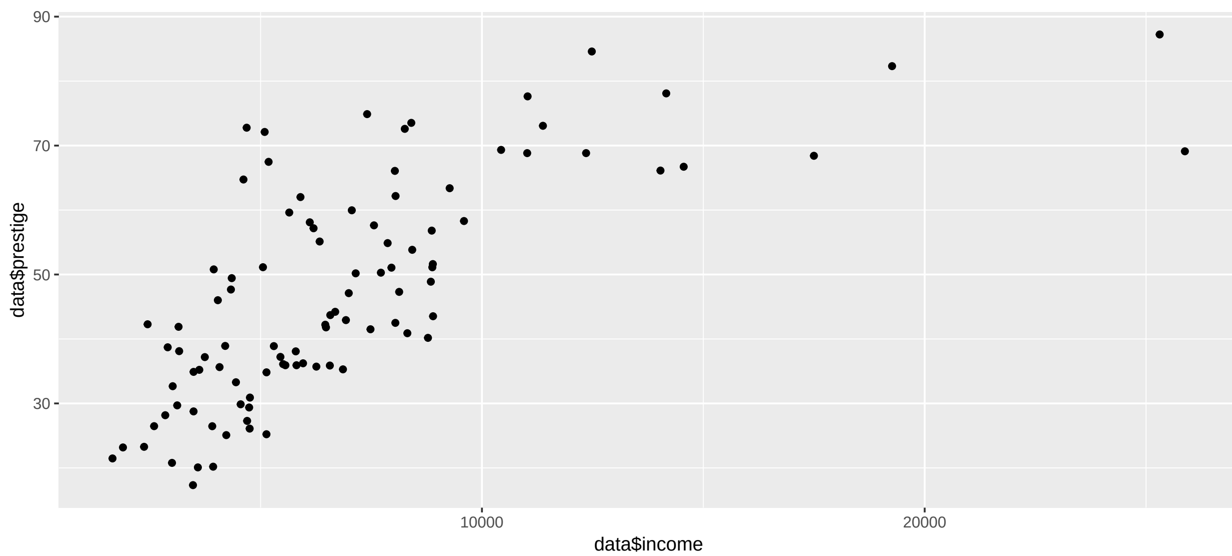

```
ggplot(data=data,aes(x=data$type,y=data$education))+geom_boxplot(aes(group=data$type))
```



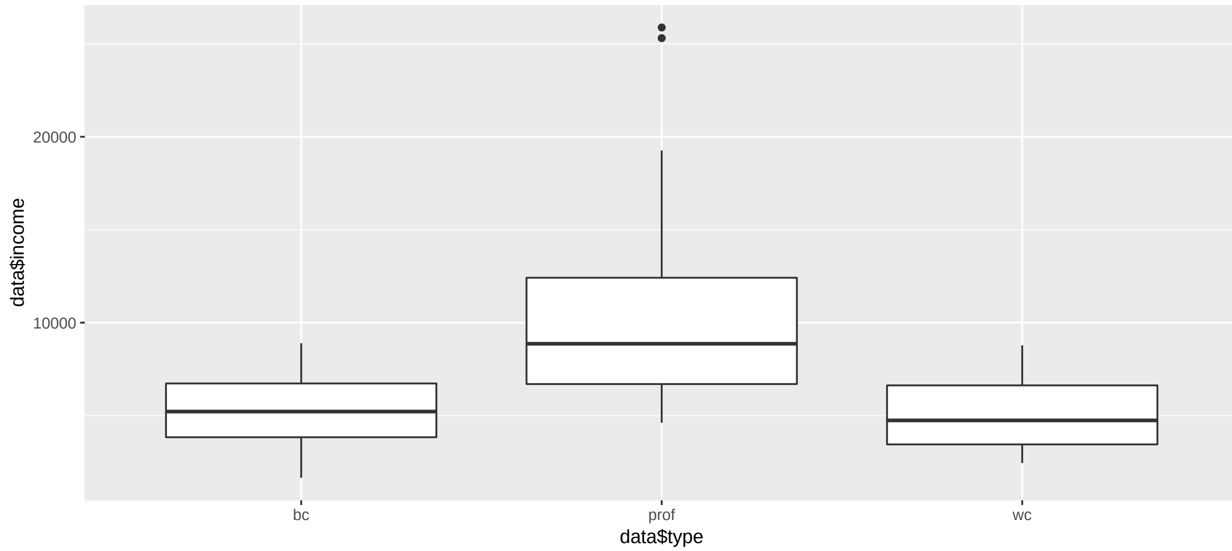

```
ggplot(data=data,aes(x=data$income,y=data$women))+geom_jitter()
```



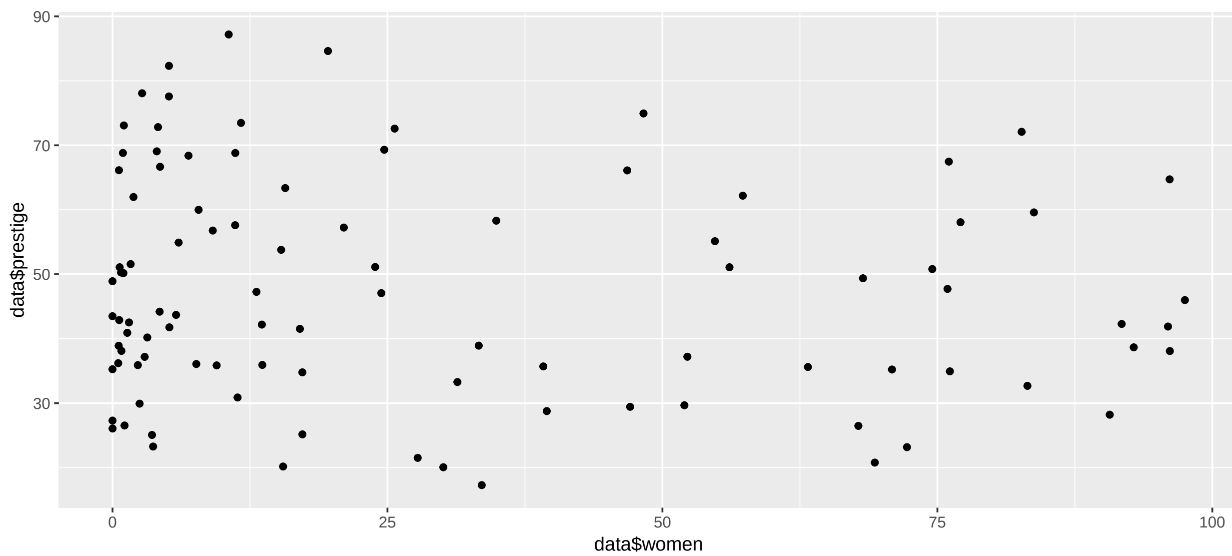

```
ggplot(data=data,aes(x=data$income,y=data$prestige))+geom_jitter()
```



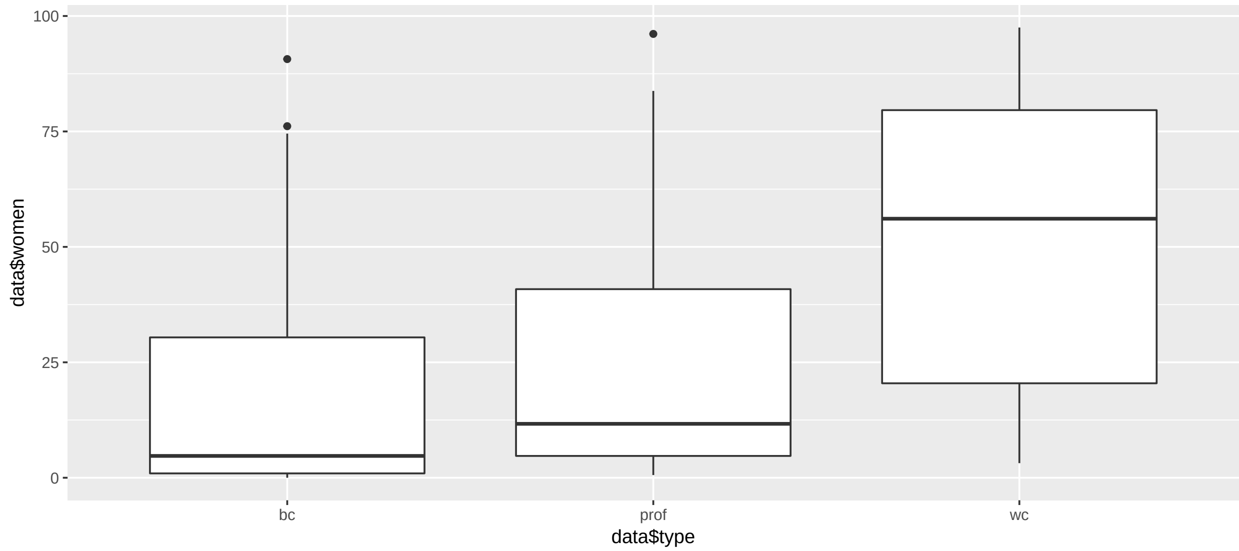

```
ggplot(data=data,aes(x=data$type,y=data$income))+geom_boxplot(aes(group=data$type))
```



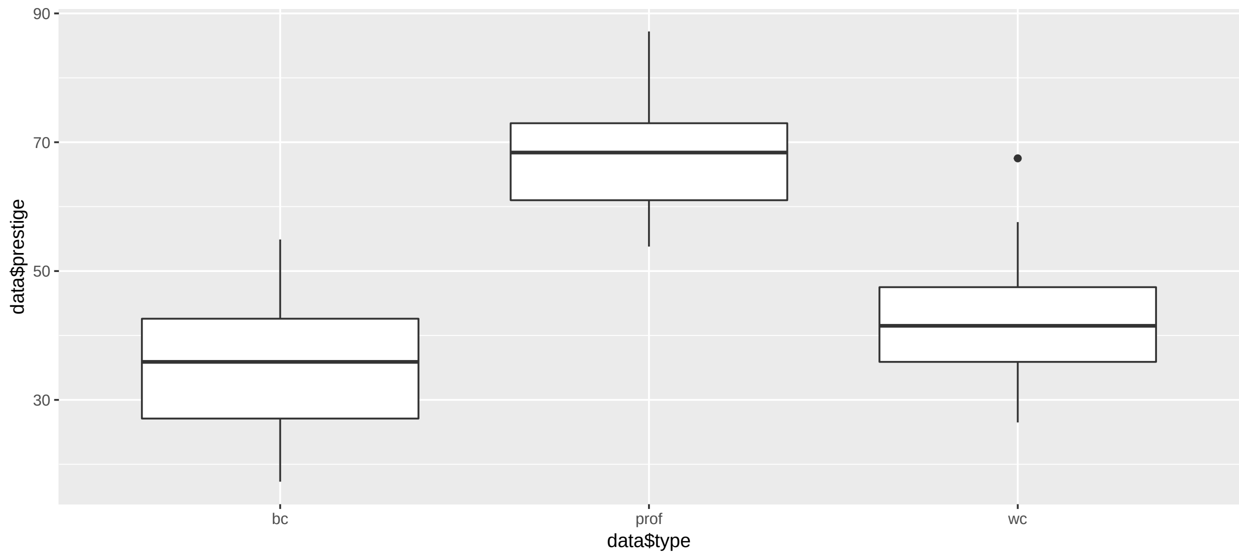

```
ggplot(data=data,aes(x=data$women,y=data$prestige))+geom_jitter()
```




```
ggplot(data=data,aes(x=data$type,y=data$women))+geom_boxplot(aes(group=data$type))
```




```r
ggplot(data=data,aes(x=data$type,y=data$prestige))+geom_boxplot(aes(group=data$type))
```




```r
data$type <- NULL
```

```r
# 1.
model1 <- lm(data = data, prestige~income)

# 2.
model2 <- lm(data = data, prestige~education)

# 3.
model3 <- lm(data = data, prestige~type1)

# 4.
model4 <- lm(data = data, prestige~women)

# 5.
model5 <- lm(data = data, prestige~education+income)
```

```r
#6.
model6 <- lm(data = data, prestige~education+type1)

#7.
model7 <- lm(data = data, prestige~education+women)

#8.
model8 <- lm(data = data, prestige~education+income+type1)

#9.
model9 <- lm(data = data, prestige~education+income+type1+women)

n_var <- dim(data)[1]

train_error <- function(n_var, model){
  return((1/n_var)*sum(residuals(model)^2))
}

opt_term <- function(n_var, p, model){
  return(((2*var(residuals(model)))/n_var) * p)
}

test_error_exp <- function(opt_term, train_error){
  return(opt_term + train_error)
}

results <- matrix(NA, ncol = 5, nrow = 9)
colnames(results) <- c("N_Param", "Trainingsfehler", "Optimismusterm",  "Erwarteter_Testfehler", "AIC")
rownames(results) <- c("model1", "model2", "model3","model4", "model5", "model6", "model7", "model8", "m
```

```r
model_list <- list(model1, model2, model3, model4, model5, model6, model7, model8, model9)
for (model in 1:length(model_list)){
  p <- length(model_list[[model]]$coefficients)
  results[model,1] <- length(model_list[[model]]$coefficients)
  results[model,2] <- train_error(n_var, model_list[[model]])
  results[model,3] <- opt_term(n_var, p, model_list[[model]])
  results[model,4] <- test_error_exp(results[model,3], results[model,2])
  results[model,5] <- AIC(model_list[[model]])
}

results
#>        N_Param Trainingsfehler Optimismusterm Erwarteter_Testfehler
#> model1       2       146.17611       6.027881             152.20399
#> model2       2        72.08576       2.972609              75.05837
#> model3       2       262.83293      10.838471             273.67140
#> model4       2       285.74478      11.783290             297.52807
#> model5       3        53.80040       3.327860              57.12826
#> model6       3        62.76835       3.882578              66.65093
#> model7       3        64.29518       3.977021              68.27220
#> model8       4        50.66315       4.178404              54.84155
#> model9       5        50.59325       5.215799              55.80905
#>             AIC
#> model1 772.6235
```

```
#> model2 703.3419
#> model3 830.1208
#> model4 838.3117
#> model5 676.6695
#> model6 691.7781
#> model7 694.1334
#> model8 672.7814
#> model9 674.6461
```

- Länge der Ausbildung scheint den größten Einfluss auf die abhängige Variable zu haben, da dass Modell 4 das beste Modell mit nur einer unabhängigen Variable ist.
- Modell 5 bietet aufbauend auf der Variablen Länge der Ausbildung die höchste Genaugkeit, durch Hinzunahme der Variable Einkommen. Frauenanteil (Modell 7) und Berufsklasse (Modell 6) sind dagegen laut AIC schlechtere Modelle.
- Weitere Hinzunahme der unabhängigen Variablen Berufsklasse und Frauenanteil führt zu keiner Verbesserung der Modellgüte.
- Modell 5 scheint das robusteste Modell zu sein, in dem nur signifikante unabhängige Variablen vorkommen. Das Risiko für Overfitting ist klein.

b)

Um AIC-Werte miteinander vergleichen zu können, müssen die Modelle auf den gleichen Daten trainiert worden sein.