# Example Regression Analysis: Statistics Lecture Exam Data

Susumu Shikano

Last compiled at 18. Juli 2022

We read the data of from the statistics lecture in the past. Note that this data is based on the real data of the statistics lecture, but it is a random sample.

```
# Setting the working directory where the data file is stored.
load(file="data/Statistics_Exam_Anonymous.RData")
```

We first create the binary variable whether individual students passed the exam.

```
exam.anonym$passed <- ifelse(exam.anonym$points>=50,1,0)

table(exam.anonym$passed)
```

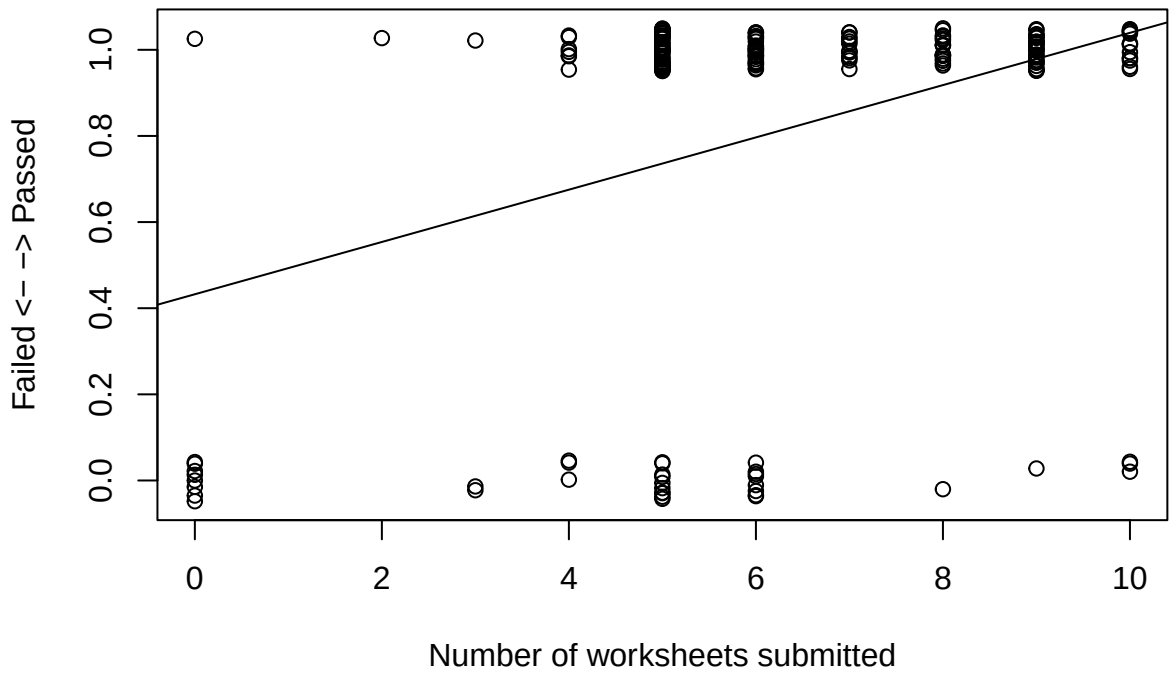
```
##
##   0   1
##  36 164
```

We regress the created binary variable on the number of worksheets submitted.

```
summary(lpm.out <- lm(passed ~ worksheet_submitted,data=exam.anonym))
```

```
##
## Call:
## lm(formula = passed ~ worksheet_submitted, data = exam.anonym)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03946  0.00607  0.08196  0.26408  0.56761
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.43239    0.07350   5.882 1.70e-08 ***
## worksheet_submitted  0.06071    0.01081   5.618 6.48e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3586 on 198 degrees of freedom
## Multiple R-squared:  0.1375, Adjusted R-squared:  0.1331
## F-statistic: 31.56 on 1 and 198 DF,  p-value: 6.48e-08
```

```
plot(jitter(exam.anonym$passed,amount=0.05) ~ exam.anonym$worksheet_submitted,
     xlab="Number of worksheets submitted",ylab="Failed <- -> Passed")
abline(lpm.out)
```




Note that our dependent variable can take only two possible values (0 or 1). How can we interpret this result?

Under the zero conditional mean assumption:

$E(y|x) = \beta_0 + \beta_1 no.worksheets$

This expected value can be interpreted as $\Pr(y = 1|x)$.

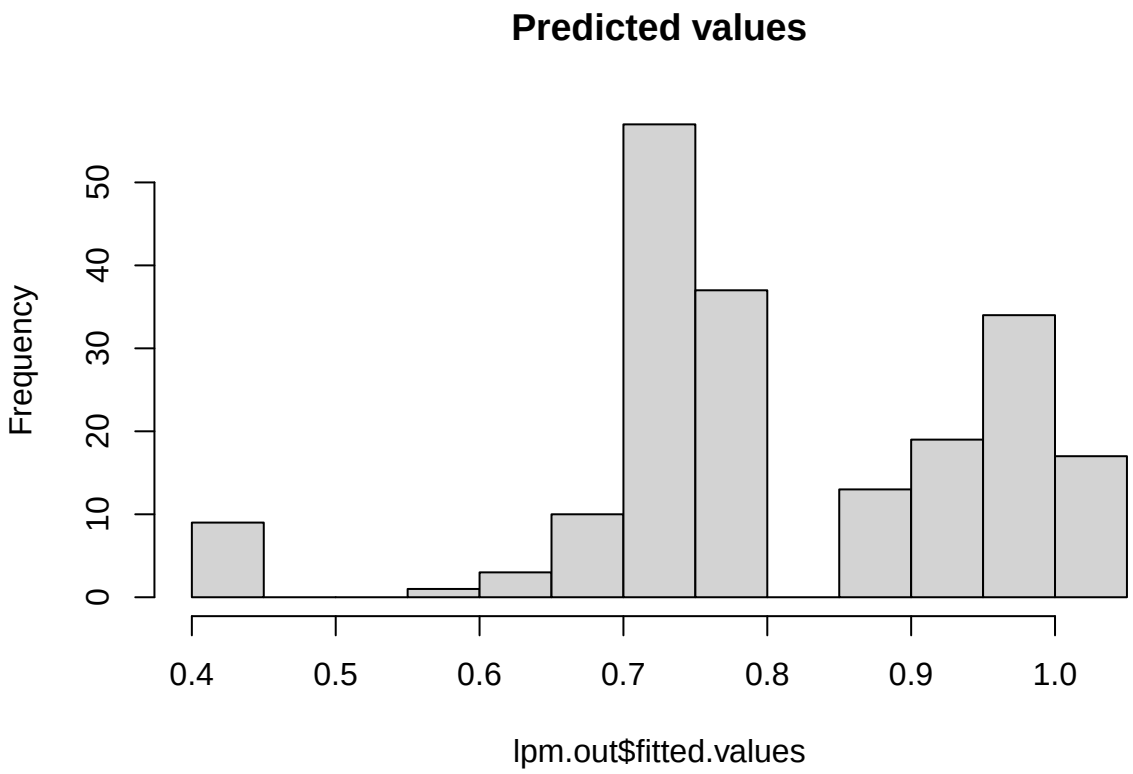
Therefore: $\Delta \Pr(y = 1|x) = \beta_1 \Delta no.worksheets$.

There are two potential drawbacks:

- Predicted value for y can be larger than 1 or smaller than 0.
- The constant linear probability change depending on x may be problematic.
- Violation of the homoskedasticity assumption (for more details see Chapter 8).

The first problem is the case in the above example:

```
hist(lpm.out$fitted.values,main="Predicted values")
```




For some respondents, we predicted values larger than 1.

To avoid this problem, we can make the prediction based on the following rule:

$\tilde{y} = 1$ if $\hat{y} \geq 0.5$ and $\tilde{y} = 0$ otherwise. In the above example:

```
predicted.y <- ifelse(lpm.out$fitted.values>=0.5,1,0)
observed.y <- exam.anonym$passed

mean(predicted.y==observed.y)
```

```
## [1] 0.855
```

We correctly predicted 85.5% of observations (**percent correctly predicted**). However, we look at the contingency table of the observed and predicted values:

```
table(predicted.y,observed.y)
```

```
##            observed.y
## predicted.y   0   1
##           0   8   1
##           1  28 163
```

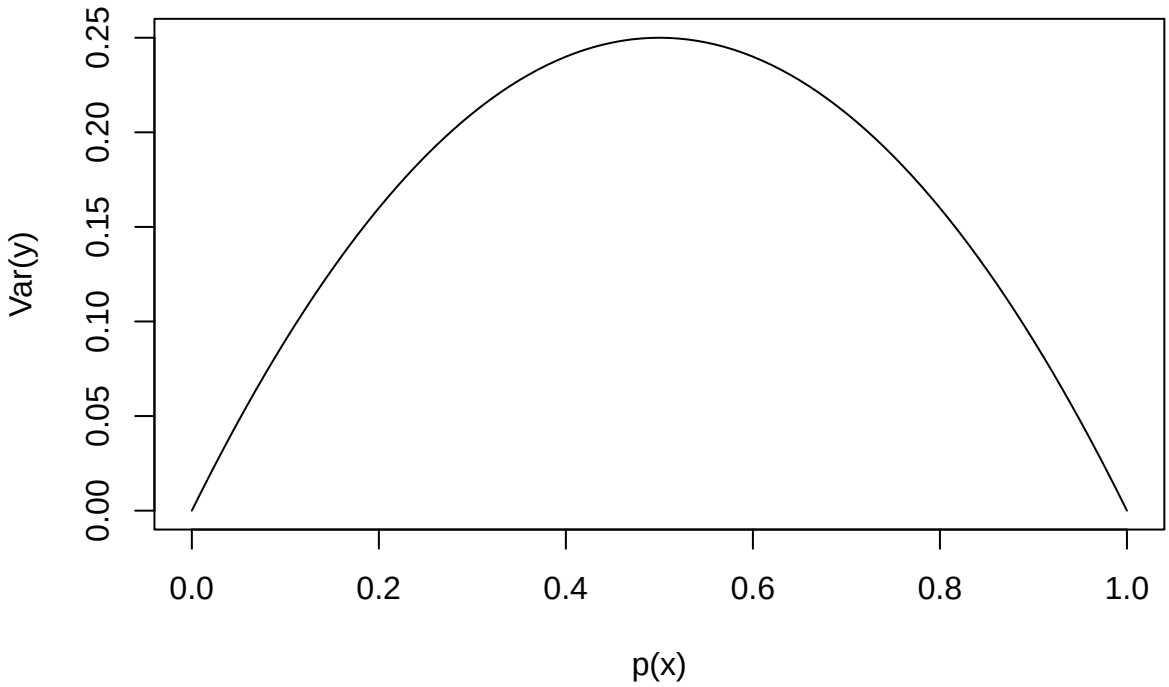
Most of the respondents (95.5%) were predicted to pass (i.e. $\hat{y} = 1$).

## Heteroskedasticity

The linear probability model must contain heteroskedasticity (except $\beta_j = 0$ for all j):

$Var(y|x) = p(x)[1 - p(x)]$ with $p(x) = \beta_0 + \beta_1 x_1 + ... + \beta_k x_k$.

```r
var.function <- function(p) p*(1-p)
curve(var.function,0,1,xlab="p(x)",ylab="Var(y)")
```




Two possible remedies:

- OLS-estimates + robust standard errors.
- WLS-estimates with the weights $\hat{h} = \hat{y}(1 - \hat{y})$.

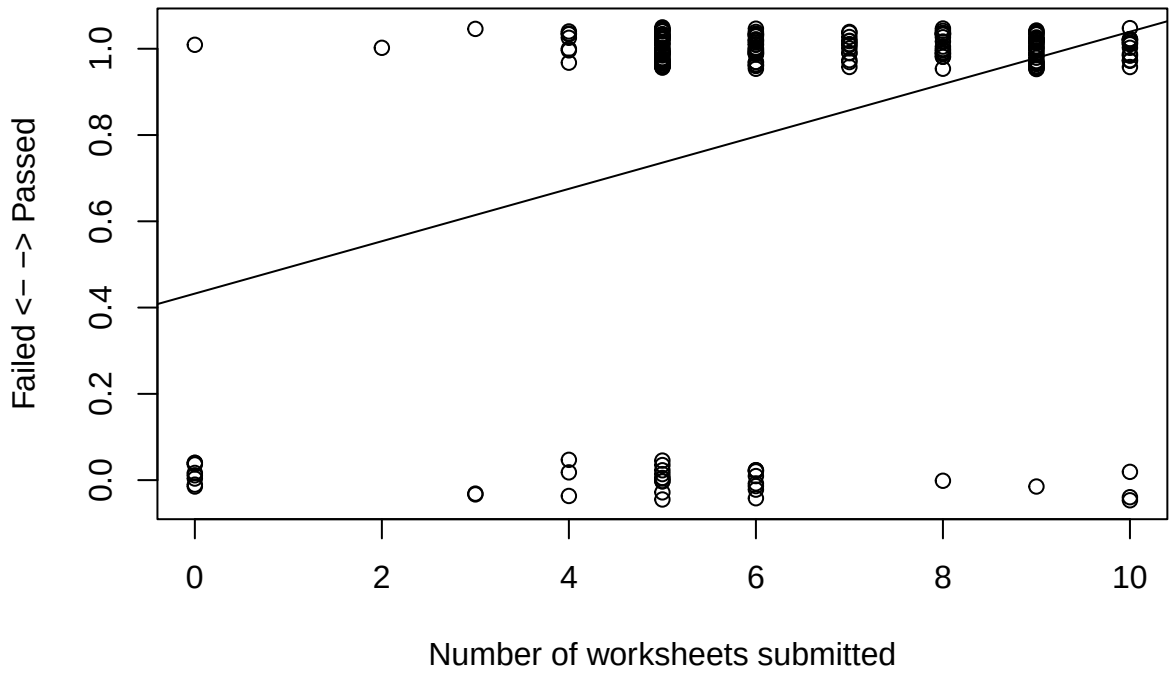
We can apply these remedies to the LPM estimated above:

```r
summary(lpm.out)
```

```
##
## Call:
## lm(formula = passed ~ worksheet_submitted, data = exam.anonym)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03946  0.00607  0.08196  0.26408  0.56761
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.43239    0.07350   5.882 1.70e-08 ***
## worksheet_submitted  0.06071    0.01081   5.618 6.48e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3586 on 198 degrees of freedom
```

```
## Multiple R-squared:  0.1375, Adjusted R-squared:  0.1331
## F-statistic: 31.56 on 1 and 198 DF,  p-value: 6.48e-08
```

```
plot(jitter(exam.anonym$passed,amount=0.05) ~ exam.anonym$worksheet_submitted,
     xlab="Number of worksheets submitted",ylab="Failed <- -> Passed")
abline(lpm.out)
```




The (heteroskedasticity-) robust standard error is:

```
var.slope.lpm.het <- sum((exam.anonym$worksheet_submitted - mean(exam.anonym$worksheet_submitted))^2*(lp

var.slope.lpm.het
```

```
## [1] 0.0001399789
```

```
sqrt(var.slope.lpm.het)
```

```
## [1] 0.01183127
```

The WLS estimates are:

```
predicted <- lpm.out$fitted.values
h.hat <- predicted * (1-predicted)

wls.lpm.out <- lm(passed ~ worksheet_submitted,data=exam.anonym,
                  weights = 1/h.hat)
```

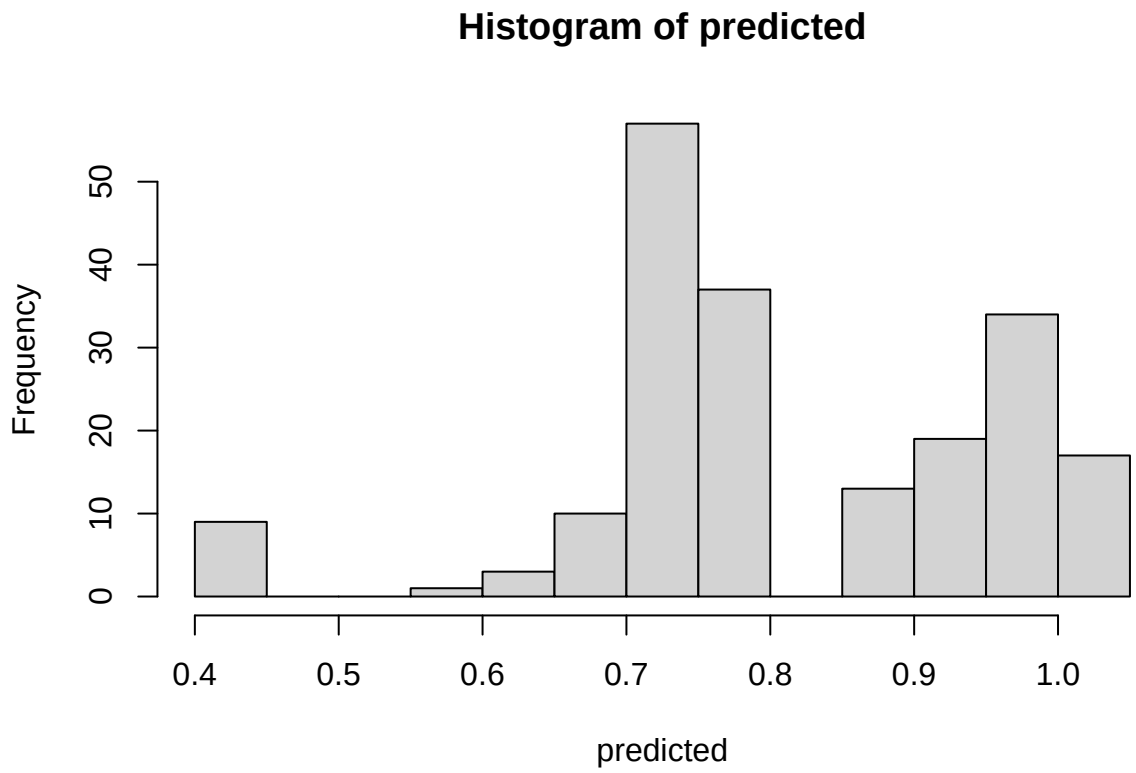
```
## Error in lm.wfit(x, y, w, offset = offset, singular.ok = singular.ok, : missing or negative weights n
```

```
summary(wls.lpm.out)
```

```
## Error in summary(wls.lpm.out): Objekt 'wls.lpm.out' nicht gefunden
```

This does not work since some predicted values are not in the range $(0, 1)$.

```
predicted <- lpm.out$fitted.values
hist(predicted)
```




The figure shows that some predictions exceeds 1. This makes $\hat{h}$ negative values, which cannot be used as weight.

One possibility is to adjust those predicted values. We can replace the predicted values above 1 with 0.99:

```
predicted <- lpm.out$fitted.values
predicted[predicted>=1] <- 0.99

h.hat <- predicted * (1-predicted)

wls.lpm.out <- lm(passed ~ worksheet_submitted,data=exam.anonym,
                  weights = 1/h.hat)
summary(wls.lpm.out)
```

```
##
## Call:
## lm(formula = passed ~ worksheet_submitted, data = exam.anonym,
##     weights = 1/h.hat)
##
```

```
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1552  0.4594  0.4738  0.5065  0.8951
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.67131    0.11510   5.833 2.19e-08 ***
## worksheet_submitted  0.02396    0.01295   1.850   0.0658 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.507 on 198 degrees of freedom
## Multiple R-squared:  0.01699,    Adjusted R-squared:  0.01202
## F-statistic: 3.422 on 1 and 198 DF,  p-value: 0.06584
```

The replaced value 0.99 was arbitrarily chosen. We can also use 0.999:

```
predicted <- lpm.out$fitted.values
predicted[predicted>=1] <- 0.999

h.hat <- predicted * (1-predicted)

wls.lpm.out <- lm(passed ~ worksheet_submitted,data=exam.anonym,
                  weights = 1/h.hat)
summary(wls.lpm.out)
```

```
##
## Call:
## lm(formula = passed ~ worksheet_submitted, data = exam.anonym,
##     weights = 1/h.hat)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -26.4859   0.4156   0.4451   0.6234   5.1527
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.796421   0.259969   3.064  0.00249 **
## worksheet_submitted 0.004072   0.026623   0.153  0.87860
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.687 on 198 degrees of freedom
## Multiple R-squared:  0.0001181,  Adjusted R-squared:  -0.004932
## F-statistic: 0.02339 on 1 and 198 DF,  p-value: 0.8786
```

Here, the result became completely different since we have a significant amount of predicted values outside of $(0, 1)$. In such cases, we should better use the heteroskedasticity-robust statistics.

## Logit and Probit Models for Binary Response

We first regress the dummy variable (pass/fail) on the number of submitted worksheets and the number of persons with whom individual students prepared the exam together:

```
exam.anonym$groupwork[is.na(exam.anonym$groupwork)] <- 0
```

```
summary(lpm.out <- lm(passed ~ worksheet_submitted+groupwork,data=exam.anonym))
```

```
##
## Call:
## lm(formula = passed ~ worksheet_submitted + groupwork, data = exam.anonym)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04221 -0.04947  0.09397  0.23015  0.62980
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.37020    0.07362   5.029 1.11e-06 ***
## worksheet_submitted  0.06028    0.01051   5.737 3.58e-08 ***
## groupwork            0.03795    0.01077   3.525 0.000526 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3487 on 197 degrees of freedom
## Multiple R-squared:  0.1887, Adjusted R-squared:  0.1804
## F-statistic: 22.91 on 2 and 197 DF,  p-value: 1.137e-09
```

Now, we estimate the logit model with the same variables:

```
logit.out <- glm(passed ~ worksheet_submitted+groupwork,
                 family=binomial(link="logit"),
                 data=exam.anonym)
summary(logit.out)
```

```
##
## Call:
## glm(formula = passed ~ worksheet_submitted + groupwork, family = binomial(link = "logit"),
##     data = exam.anonym)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7656   0.2312   0.3843   0.6273   1.7577
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.30488    0.57610  -2.265  0.02351 *
## worksheet_submitted  0.41118    0.09772   4.208 2.58e-05 ***
## groupwork            0.36352    0.13676   2.658  0.00786 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188.56  on 199  degrees of freedom
## Residual deviance: 150.37  on 197  degrees of freedom
## AIC: 156.37
##
## Number of Fisher Scoring iterations: 5
```

and the probit model:

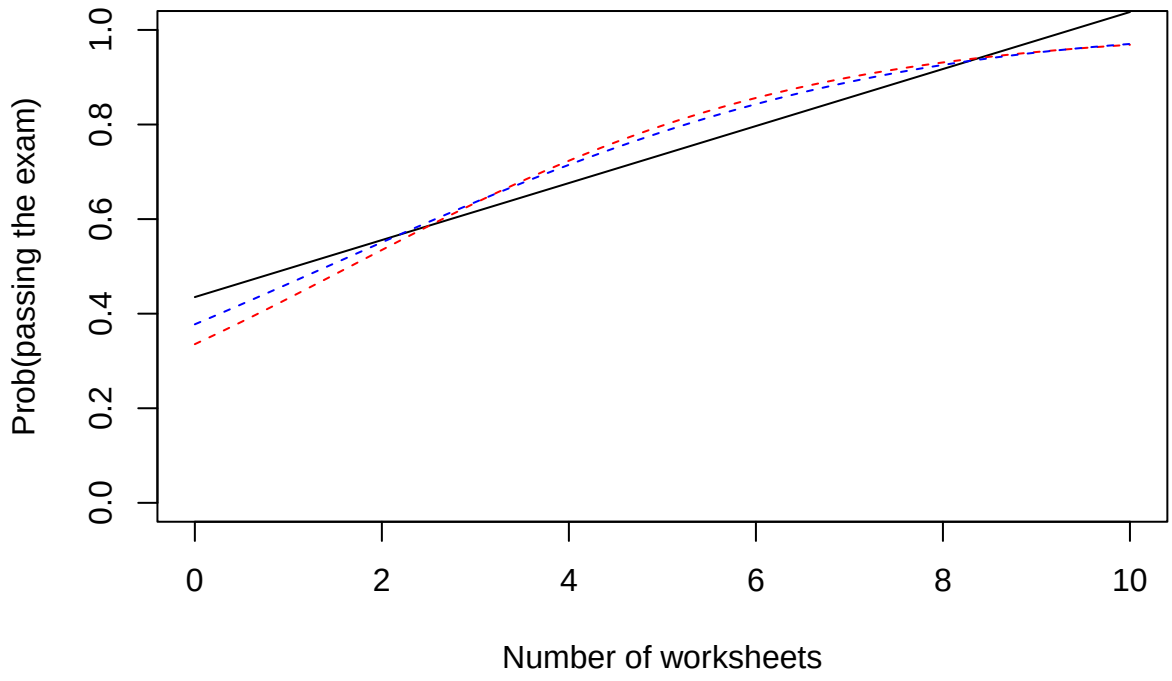
```r
probit.out <- glm(passed ~ worksheet_submitted+groupwork,
                  family=binomial(link="probit"),
                  data=exam.anonym)
summary(probit.out)
```

```
##
## Call:
## glm(formula = passed ~ worksheet_submitted + groupwork, family = binomial(link = "probit"),
##     data = exam.anonym)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8183   0.2195   0.4030   0.6530   1.6391
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.64030    0.31637  -2.024   0.0430 *
## worksheet_submitted  0.21982    0.05091   4.318 1.57e-05 ***
## groupwork            0.19199    0.06688   2.870   0.0041 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188.56  on 199  degrees of freedom
## Residual deviance: 152.04  on 197  degrees of freedom
## AIC: 158.04
##
## Number of Fisher Scoring iterations: 6
```

We compare the predicted values based on three different models. Below, we predict the probability of passing the exam for different number of submitted worksheets, while keeping the other variable (groupwork) being the average:

```r
# average values for groupwork
groupwork.bar <- mean(exam.anonym$groupwork)

# generate different values for the nuber of submitted worksheets between its minimum and maximum value
worksheets.values <- seq(0,10,length=100)

# predict for different catholic shares based on different model
## LPM
predict.lpm <- coefficients(lpm.out)[1] +
               coefficients(lpm.out)[2] *worksheets.values +
               coefficients(lpm.out)[3] *groupwork.bar

predict.logit <- coefficients(logit.out)[1] +
               coefficients(logit.out)[2] *worksheets.values +
               coefficients(logit.out)[3] *groupwork.bar
predict.logit <- exp(predict.logit)/(1+exp(predict.logit))

predict.probit <- coefficients(probit.out)[1] +
               coefficients(probit.out)[2] *worksheets.values +
               coefficients(probit.out)[3] *groupwork.bar
predict.probit <- pnorm(predict.probit)
```

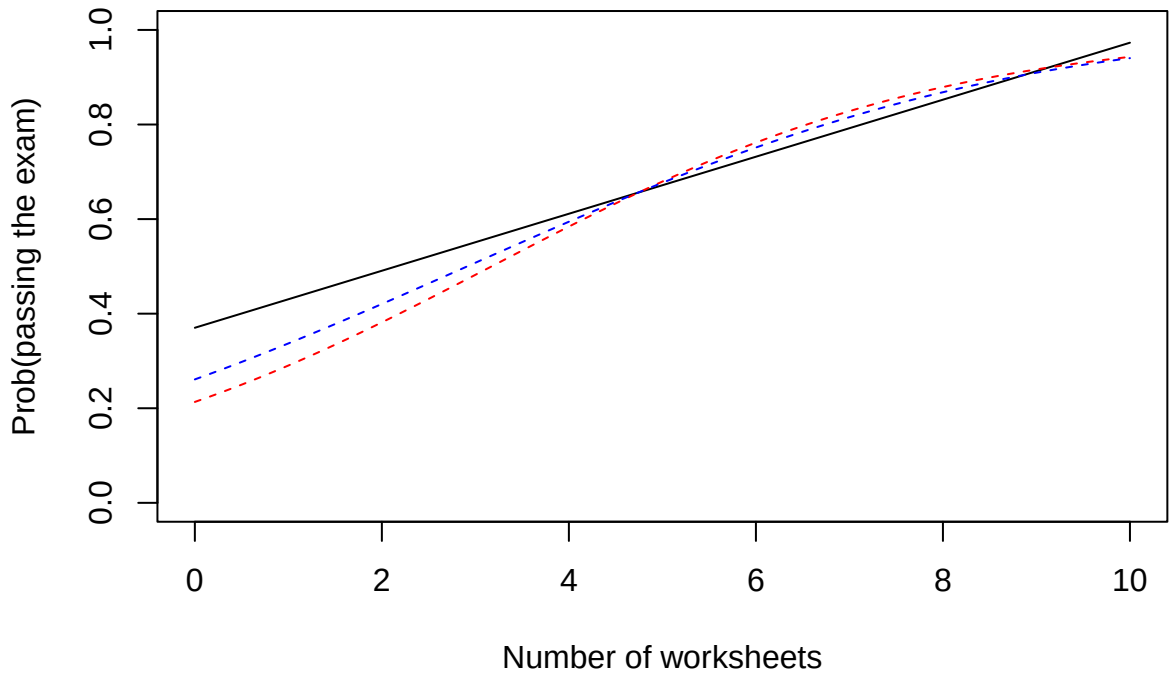
```
plot(worksheets.values,predict.lpm,ylim=c(0,1),type="l",
     xlab="Number of worksheets",ylab="Prob(passing the exam)")
lines(worksheets.values, predict.logit, col="red",lty=2)
lines(worksheets.values, predict.probit, col="blue",lty=2)
```




Now, we change the average value for groupwork (1.71) variable to the minimum value (0):

```
# average values for GDP, but the maximum value for abi
groupwork.min <- 0


# predict for different catholic shares based on different model
## LPM
predict.lpm <- coefficients(lpm.out)[1] +
               coefficients(lpm.out)[2] *worksheets.values +
               coefficients(lpm.out)[3] *groupwork.min

predict.logit <- coefficients(logit.out)[1] +
               coefficients(logit.out)[2] *worksheets.values +
               coefficients(logit.out)[3] *groupwork.min
predict.logit <- exp(predict.logit)/(1+exp(predict.logit))

predict.probit <- coefficients(probit.out)[1] +
               coefficients(probit.out)[2] *worksheets.values +
               coefficients(probit.out)[3] *groupwork.min
predict.probit <- pnorm(predict.probit)
```

```
plot(worksheets.values,predict.lpm,ylim=c(0,1),type="l",
     xlab="Number of worksheets",ylab="Prob(passing the exam)")
lines(worksheets.values, predict.logit, col="red",lty=2)
lines(worksheets.values, predict.probit, col="blue",lty=2)
```




It is apparent that the partial effect of the same number of submitted worksheets differ depending on the size of learning group.

### 17.1 Logit and Probit Models for Binary Response: LR-test

We focus here only on the probit model:

```
probit.out <- glm(passed ~ worksheet_submitted+groupwork,
                  family=binomial(link="probit"),
                  data=exam.anonym)
summary(probit.out)
```

```
##
## Call:
## glm(formula = passed ~ worksheet_submitted + groupwork, family = binomial(link = "probit"),
##     data = exam.anonym)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8183   0.2195   0.4030   0.6530   1.6391
##
```

```
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.64030    0.31637  -2.024   0.0430 *
## worksheet_submitted  0.21982    0.05091   4.318 1.57e-05 ***
## groupwork            0.19199    0.06688   2.870   0.0041 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188.56  on 199  degrees of freedom
## Residual deviance: 152.04  on 197  degrees of freedom
## AIC: 158.04
##
## Number of Fisher Scoring iterations: 6
```

Concerning this model, we test whether all the included independent variables are relevant as a whole to explain the probability to pass the exam. To do this, we can estimate the restricted model without any independent variables:

```
probit.out.res <- glm(passed ~ 1,family=binomial(link="probit"),
                      data=exam.anonym)
summary(probit.out.res)
```

```
##
## Call:
## glm(formula = passed ~ 1, family = binomial(link = "probit"),
##     data = exam.anonym)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.852   0.630   0.630   0.630   0.630
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.9154     0.1035   8.842   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 188.56  on 199  degrees of freedom
## Residual deviance: 188.56  on 199  degrees of freedom
## AIC: 190.56
##
## Number of Fisher Scoring iterations: 4
```

We can obtain the log likelihood of both models:

```
logLik(probit.out)
```

```
## 'log Lik.' -76.01925 (df=3)
```

```
logLik(probit.out.res)
```

```
## 'log Lik.' -94.2787 (df=1)
```

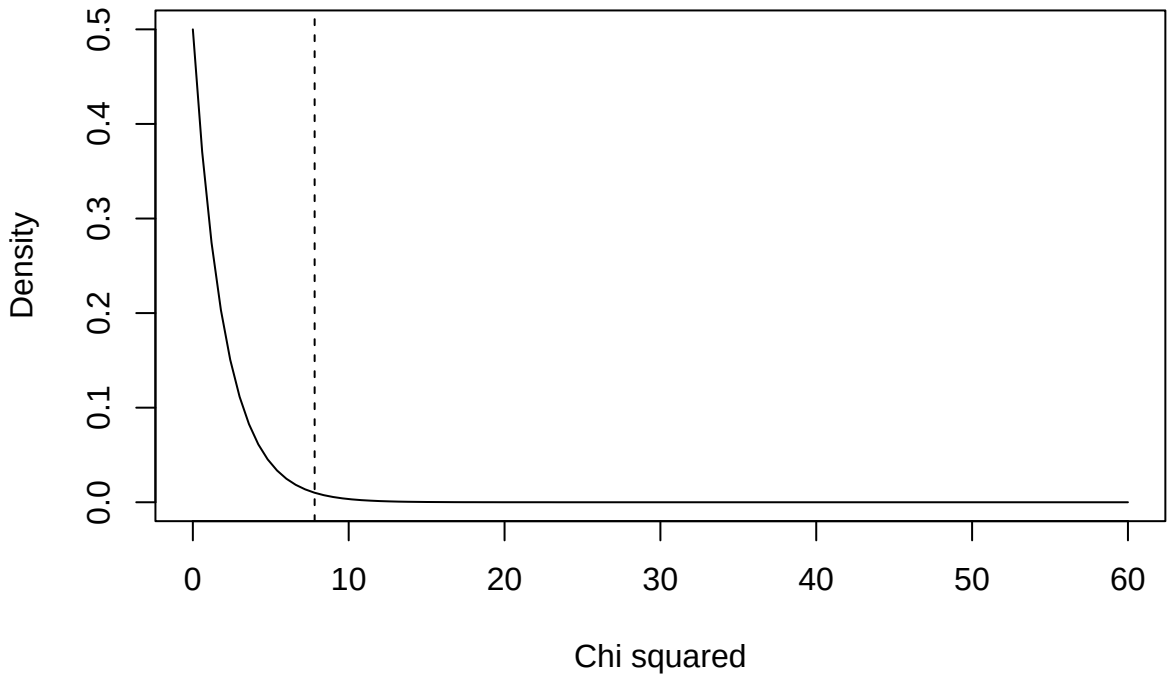
. . . and the likelihood ration, as well:

```
LR <- as.numeric(2 * (logLik(probit.out) - logLik(probit.out.res)))
LR
```

```
## [1] 36.51889
```

This value follows the chi-square distribution with 2 df:

```
chisq.func <- function(x) dchisq(x,df=2)

curve(chisq.func,0,60,ylab="Density",xlab="Chi squared")
abline(v=qchisq(0.95,df=3),lty=2)
```




At the significance level of 5%, we can reject the Null-hypothesis, "the restricted and unrestricted models do not differ".

## 17.3 The Poisson Regression Model

Suppose we are interested in the number of submitted worksheets and investigate its relationship with the points, which the students earned at the method exam in the previous semester.

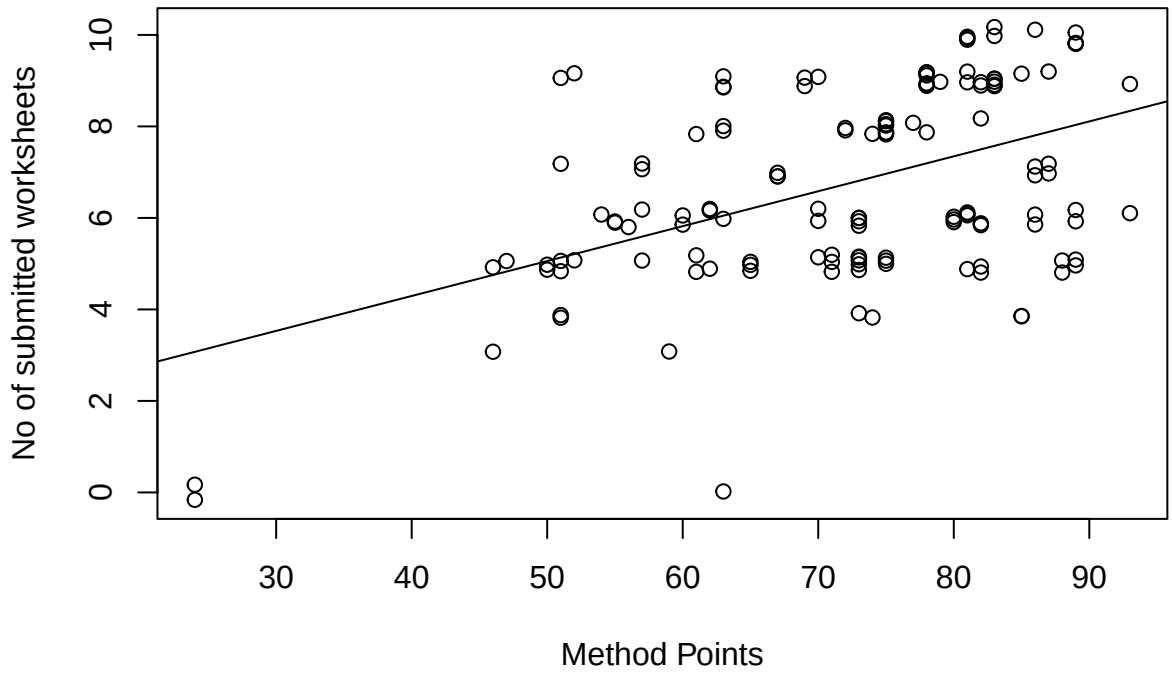
```
lm.out <- lm(worksheet_submitted ~ method.points,data=exam.anonym)
summary(lm.out)
```

```
##
## Call:
## lm(formula = worksheet_submitted ~ method.points, data = exam.anonym)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -6.0488 -1.4604  0.1801  1.4633  3.8668
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.24180    0.87758   1.415    0.159
## method.points  0.07630    0.01198   6.369 2.67e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.854 on 137 degrees of freedom
##   (61 Beobachtungen als fehlend gelöscht)
## Multiple R-squared:  0.2285, Adjusted R-squared:  0.2228
## F-statistic: 40.57 on 1 and 137 DF,  p-value: 2.672e-09
```

There is a positive effect...

```
plot(jitter(exam.anonym$worksheet_submitted) ~ exam.anonym$method.points,
     ylab="No of submitted worksheets",xlab="Method Points")
abline(lm.out)
```




... as we can see in the figure here.

At the same time, we also see some potential violation of the zero conditional mean assumption.

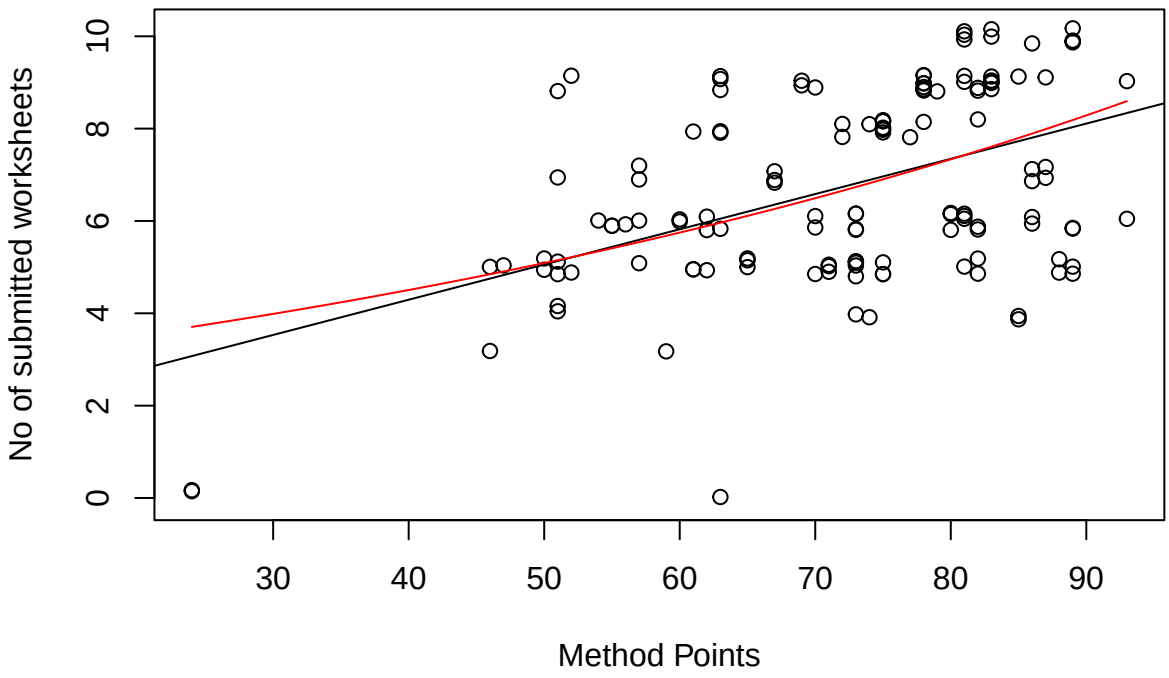
Now, we switch to the Poisson model:

```
pois.out <- glm(worksheet_submitted ~ method.points,data=exam.anonym,family=poisson)
summary(pois.out)
```

```
##
## Call:
## glm(formula = worksheet_submitted ~ method.points, family = poisson,
##     data = exam.anonym)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4531  -0.5573   0.0696   0.5420   1.5320
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.017377   0.201160   5.058 4.25e-07 ***
## method.points 0.012191   0.002681   4.546 5.46e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 110.171  on 138  degrees of freedom
## Residual deviance:  88.434  on 137  degrees of freedom
##   (61 Beobachtungen als fehlend gelöscht)
## AIC: 603.2
##
## Number of Fisher Scoring iterations: 4
```

The result shows a positive efefct of the method points. This effect is however not linear because we use the exponential function.

```
method.values <- seq(min(exam.anonym$method.points,na.rm=T),
                     max(exam.anonym$method.points,na.rm=T),length=100)
predict <- coefficients(pois.out)[1] + coefficients(pois.out)[2]*method.values
predict <- exp(predict)


plot(jitter(exam.anonym$worksheet_submitted) ~ exam.anonym$method.points,
     ylab="No of submitted worksheets",xlab="Method Points")
abline(lm.out)
lines(method.values, predict,col="red")
```




This graphic compares the predicted values of the linear regression model (black line) with those of the Poisson model (red line).