Visualizing Errors: Regression

Youmi Suk

School of Data Science, University of Virginia

3. 13. 2022

Overview

1. Linear regression with theoretical confidence intervals
2. Linear regression with bootstrap confidence intervals
3. Loess with confidence intervals
4. Binary outcome: logistic regression

Linear regression with theoretical confidence intervals

Motivating Example

- Income and Occupational Experience:

- *Example*: simulated data of monthly net income (1922 observations)

Description vs. Inference

- **Description** (of sample):
  - Data $\rightarrow$ Description
- **Statistical Inference** (from sample to target population):
  - Data + Statistical Assumptions $\rightarrow$ Conclusions about Population
- **Causal Inference** (about cause-effect relationships):
  - Data + Causal Assumptions + Statistical Assumptions $\rightarrow$ Causal Conclusions

Example: Statistical Assumptions in Linear Regression

1. Linearity: $Y = \alpha + \beta X + \epsilon$
2. X is fixed, or measured without error
3. Independence (simple random sampling)
4. Zero Conditional Mean

   $E(\epsilon|X = x) = 0 \rightarrow E(\epsilon) = 0$
5. Constant Variance (homoskedasticity)

   $Var(\epsilon|X = x) = \sigma_\epsilon^2$
6. Normality

Example: Causal Assumptions

- Neyman-Rubin's causal model (Neyman, 1923; Rubin, 1974) under the potential outcome framework
  - $Y_i(1)$ is the potential treatment outcome if treated ($T_i = 1$)
  - $Y_i(0)$ is the potential control outcome if untreated ($T_i = 0$)
  - The observed outcome, $Y_i = T_i Y_i(1) + (1 - T_i) Y_i(0)$, under the stable unit treatment value assumption (SUTVA; Rubin, 1986)
- The average treatment effect, $E(Y_i(1) - Y_i(0))$ is identified under strong ignorability assumption:
  - Unconfoundness: $(Y_i(1), Y_i(0)) \perp T_i | X_i$
  - Positivity: $0 < P(T_i = 1 | X_i) < 1$

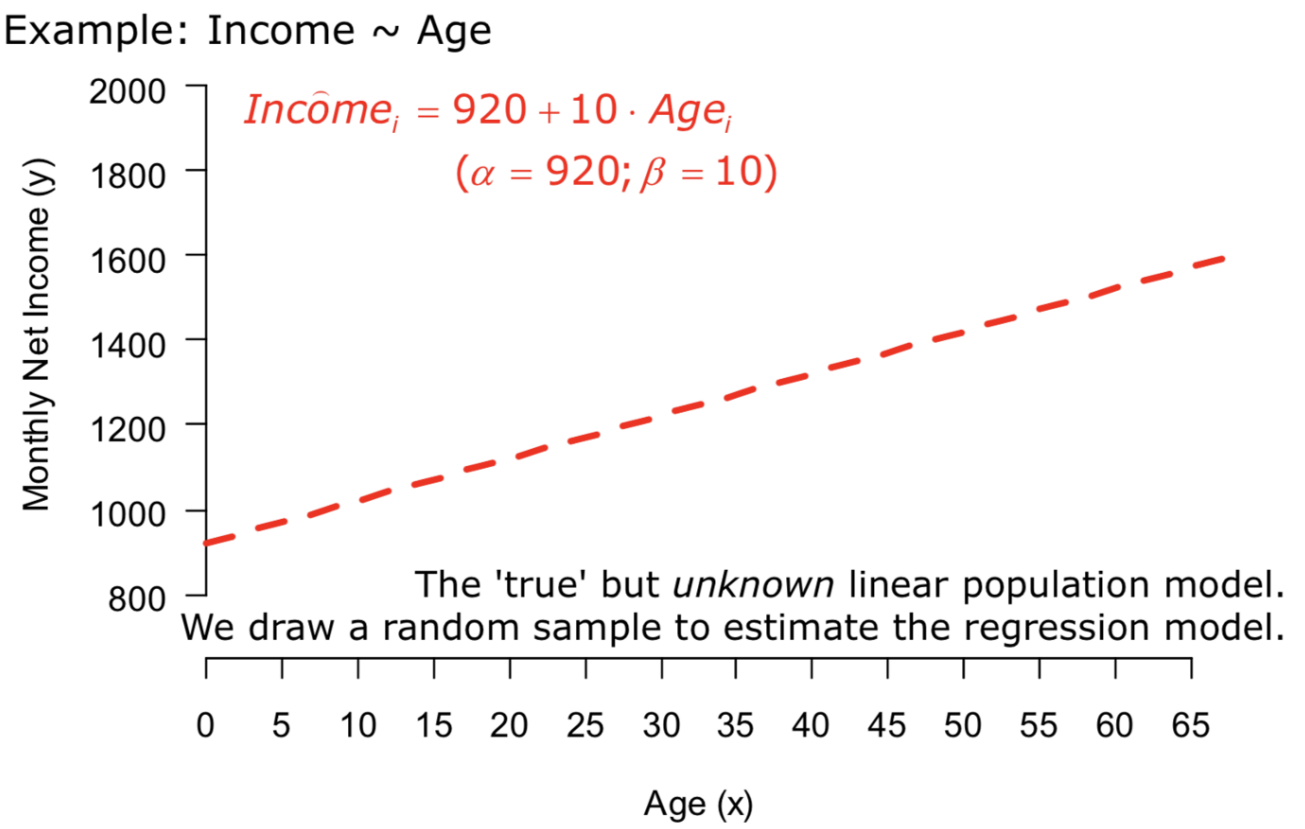
Description vs. Inference

- Data alone do not suffice to draw statistical conclusions!
- **Statistical inference** requires statistical assumptions that relate the data to the population of interest. $\rightarrow$ Without statistical assumptions, no statistical inference about the target population.
- **Causal inference** requires causal assumptions that allow us to separate causal association from non-causal association. $\rightarrow$ Without causation in, no causation out.

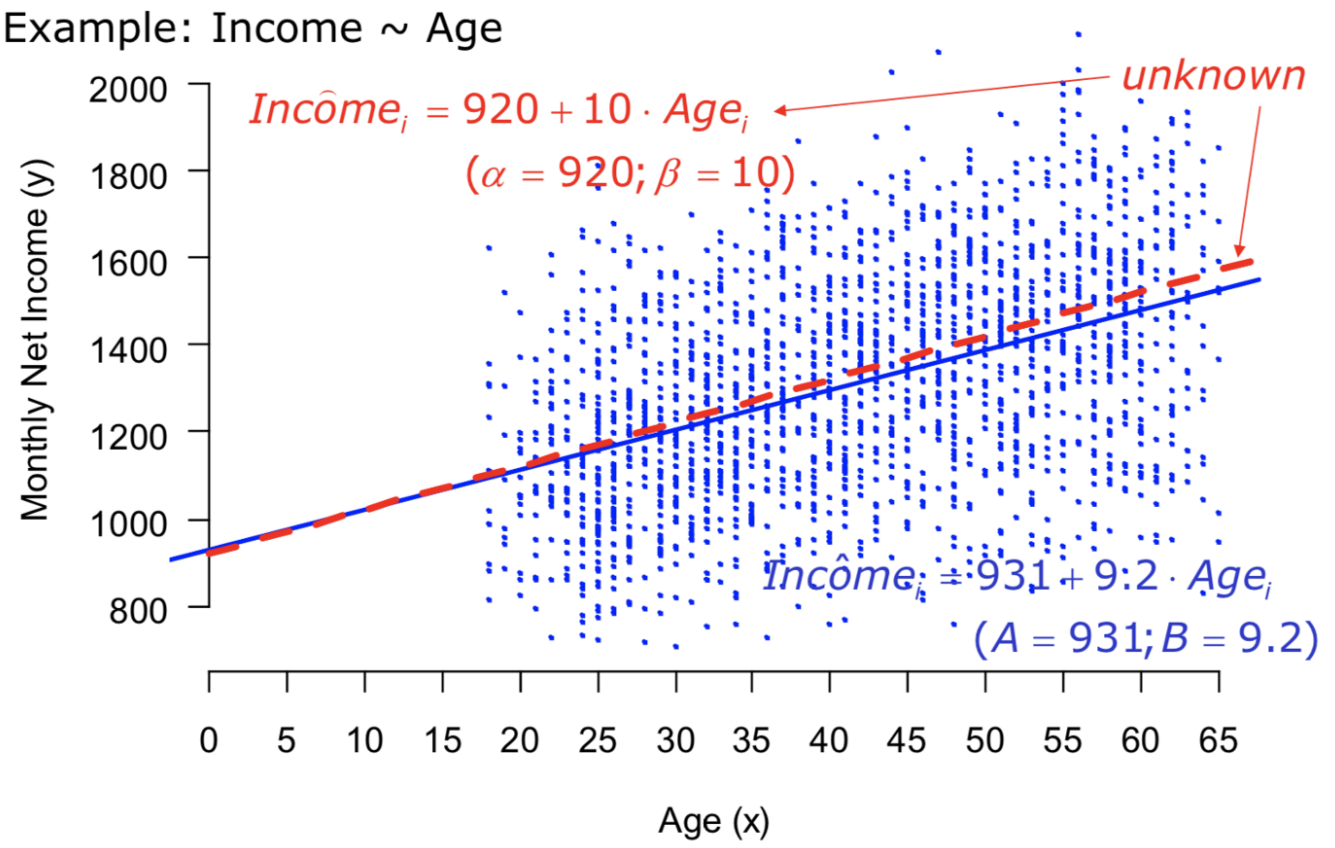
Population Model

- Everything we discussed about regression so far was purely descriptive and can be done without any assumptions.
- However, when we are interested in making inference about a well specified target population (from which a random sample was drawn), we need some assumptions.
- Note that this lecture minimizes formal notations and highlights how to visualize statistical uncertainty.

Population Model



Population Model & Estimated Model



Inference & Regression

In estimating the population model **sampling error** is involved. Estimated coefficients A and B will very likely deviate from the population parameters α and β. However, we can try to assess the uncertainty associated with our estimates A and B.

This can be done by calculating **confidence intervals for the regression coefficients**. For this purpose we have to determine the standard errors for A and B.

Also, we can determine the standard errors for the mean value of Y for a particular value of X_i (i.e., a conditional mean of Y given X) $\rightarrow$ **confidence interval of predicted means**.

R Example: 95% confidence interval of regression coefficients

```
out.lm <- lm(income ~ age, data = incex)
confint(out.lm)
```

```
##                   2.5 %    97.5 %
## (Intercept) 894.114059 967.17031
## age           8.334461  10.01613
```

R Example: 95% confidence interval of predicted means

```
predict(out.lm, data.frame(age = 30), interval = 'confidence')
```

```
##        fit      lwr     upr
## 1 1205.901 1191.682 1220.12
```

- With a confidence of 95% the interval [1192; 1220] (in EUR) covers the true mean income for the population of 30 year old employees.

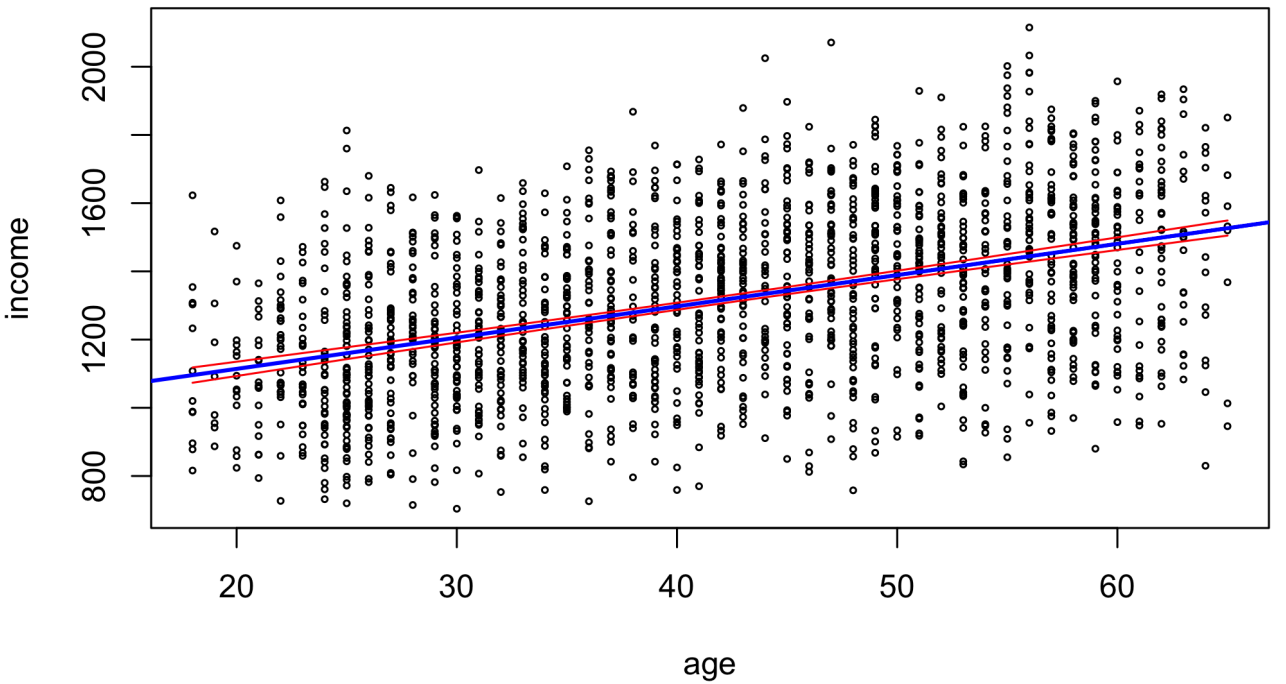
R Example: 95% confidence envelope for a linear line

```
age.pre <- 18:65
inc.pre <- predict(out.lm, data.frame(age = age.pre), interval = 'cor
kable(head(inc.pre))
```

fit	lwr	upr
1095.798	1073.385	1118.210
1104.973	1083.304	1126.641
1114.148	1093.216	1135.080
1123.323	1103.120	1143.527
1132.499	1113.014	1151.983
1141.674	1122.899	1160.449

R Example: 95% confidence envelope plot

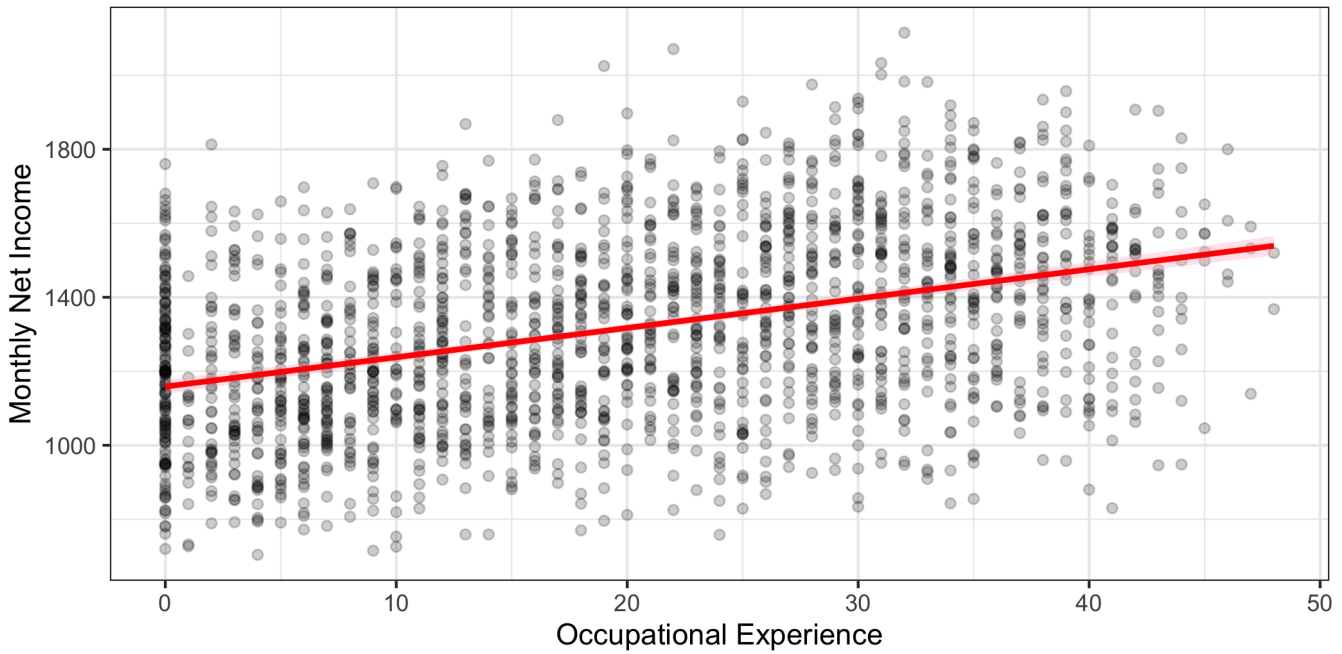
```
par(mar=c(4, 4, 0.5, 2)) # bottom, left, top, right
plot(income ~ age, data = incex, cex = .4)
abline(out.lm, col = 'blue', lwd = 2) # add linear reg.
lines(age.pre, inc.pre[, 2], col = 'red') # add lower limit
lines(age.pre, inc.pre[, 3], col = 'red') # add upper limit
```



R Example: ggplot2

- use `geom_smooth(..., se=TRUE, level=0.95)` (or `stat_smooth(..., geom = "smooth", se=TRUE, level=0.95)`)

```
ggplot(incex, aes(x=oexp, y=income)) + geom_point(alpha=0.2) + labs(x
        geom_smooth(method='lm', formula= y~x, col="red",
                    se = TRUE, level = 0.95, fill = "pink") + theme_b
```



R Example: Subset

- Draw confidence envelopes with a subset of the first 100 observations.
- For linear regression, confidence intervals are affected by sample size N, the deviation from the sample mean of X (i.e., $X - \bar{X}$), variance of X, and the sample residual variance (variance error of the estimate/regression).

```
incex2 <- incex[1:100,]
ggplot(incex2, aes(x=oexp, y=income)) + geom_point(alpha=0.2) + labs(
        geom_smooth(method='lm', formula= y~x, col="red",
                    se = TRUE, level = 0.95, fill = "pink")
```

R Example: geom_ribbon

- Add lower CIs and upper CIs manually using `geom_ribbon`

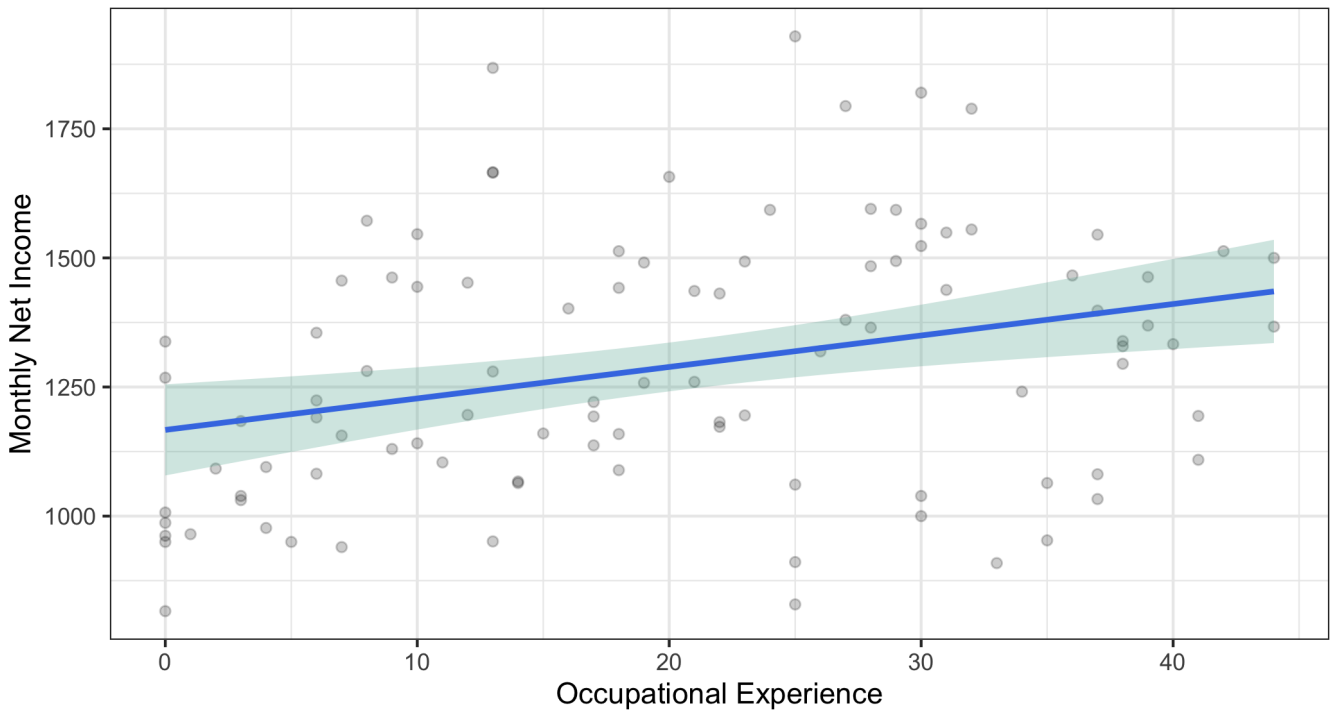
```
out.lm2 <- lm(income ~ oexp, incex2)
inc.pre2 <- data.frame(predict(out.lm2, interval = 'confidence'))
incex3 <- cbind(incex2, inc.pre2) # combine two datasets
kable(head(incex3))
```

sex	age	edu	occ	oexp	income	fit	lwr	upr
female	62	low	low	35	953	1380.246	1307.878	1452.614
male	32	high	high	6	1224	1203.481	1133.052	1273.909
male	56	med.	high	36	1466	1386.341	1311.134	1461.549
female	63	med.	med.	38	1339	1398.532	1317.440	1479.624
male	20	low	low	3	1184	1185.195	1106.191	1264.198
female	38	med.	med.	12	1196	1240.053	1184.069	1296.037

R Example: geom_ribbon

- Add lower CIs and upper CIs manually using `geom_ribbon`

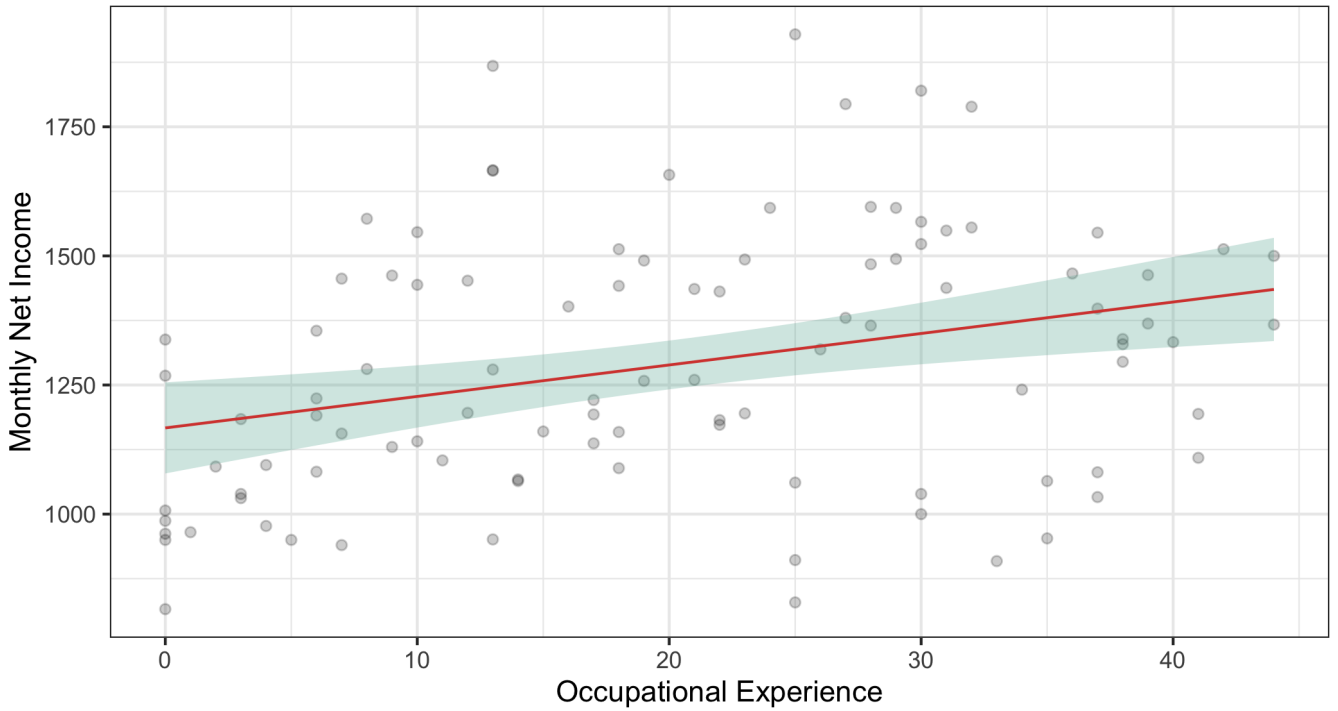
```
ggplot(incex3, aes(x=oexp, y=income)) + geom_point(alpha=0.2) + labs(
        geom_smooth(method='lm', formula= y~x, se = FALSE) +
        geom_ribbon(aes(ymin = lwr, ymax = upr), alpha = 0.3, fill="#
```



R Example: geom_ribbon + geom_line

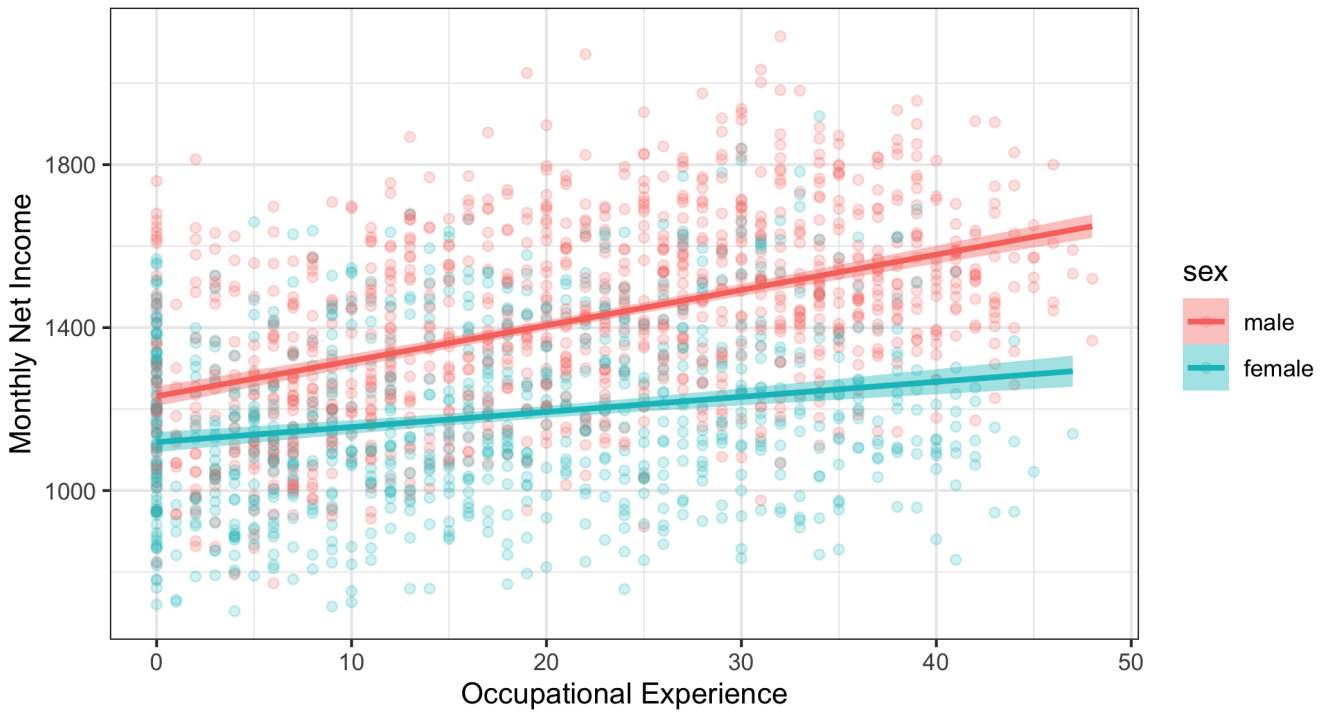
- Add CIs and the fitted line manually using `geom_ribbon` and `geom_line`

```
ggplot(incex3, aes(x=oexp, y=income)) + geom_point(alpha=0.2) + labs(
        geom_line(aes(y=fit), col="red") +
        geom_ribbon(aes(ymin = lwr, ymax = upr), alpha = 0.3, fill="#
```



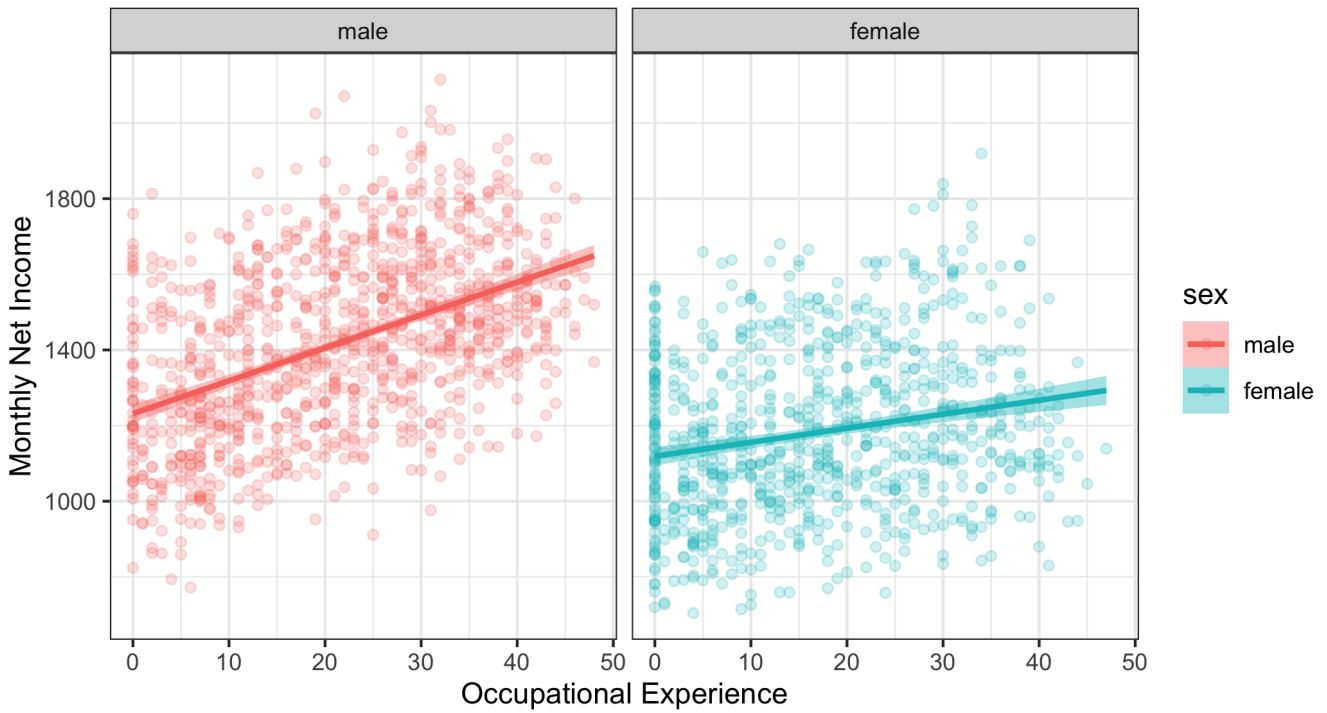
R Example: Two regression lines

```
ggplot(incex, aes(x=oexp, y=income, color=sex, fill=sex)) + geom_poir
        geom_smooth(method='lm', formula= y~x, se = TRUE) + theme_bw(
```



R Example: Two regression lines

```
ggplot(incex, aes(x=oexp, y=income, color=sex, fill=sex)) + geom_poir
        geom_smooth(method='lm', formula= y~x, se = TRUE) +
        facet_grid(~ sex) + theme_bw()
```



Linear regression with bootstrap confidence intervals

Bootstrapping

- Bootstrapping is a very general and flexible approach. It can be applied to all possible statistics of interest, but here we only consider regression coefficients and predicted values.
- The basic idea of the bootstrap is to assess a statistic's sampling distribution by mimicking the thought-experiment of repeatedly drawing a sample of the same size from the underlying target population. But instead of randomly drawing from the target population, we actually draw repeated random samples (with replacement) from the sample at hand.
- We treat the "sample" as the "population" and "(re)-sample" from that "population" over and over.

Bootstrapping

Bootstrapping procedure (basic idea):

1. Draw a bootstrap sample of size n from the sample (with replacement)

2. Use the bootstrap sample to compute the statistic of interest (e.g.,, estimate the regression model and extract the estimate b_r)

3. Repeat steps 1. & 2. R times (r = 1, …, R) and save the estimate vectors $b_1, b_2, \ldots, b_R$. From the bootstrapped sampling distribution of the R statistics, $b_1, b_2, \ldots, b_R$, compute the standard deviation (= standard error) or directly determine the confidence interval from the bootstrapped sampling distribtuion.

R Example: The `boot` package

There is a package `boot` with a function `boot()` that does the bootstrap for many situations. The function `boot()` requires three arguments: (1) the data from the original sample (here, `incex`); (2) a function to compute the statistics from the data where the first argument is the data and the second argument is the indices of the observations in the bootstrap sample; (3) the number of bootstrap replicates.

```
library(boot)
b.stat <- function(data, i)
{
   b.dat <- data[i ,]
   out.lm <- lm(income ~ oexp, b.dat)
   predict(out.lm, data.frame(oexp=incex2$oexp))
}

b.out <- boot(incex2, b.stat, R = 1000) # R = num of replications
```

```r
boot.ci(b.out, index = 1, type = 'perc') # 95% CI for the first obse
```

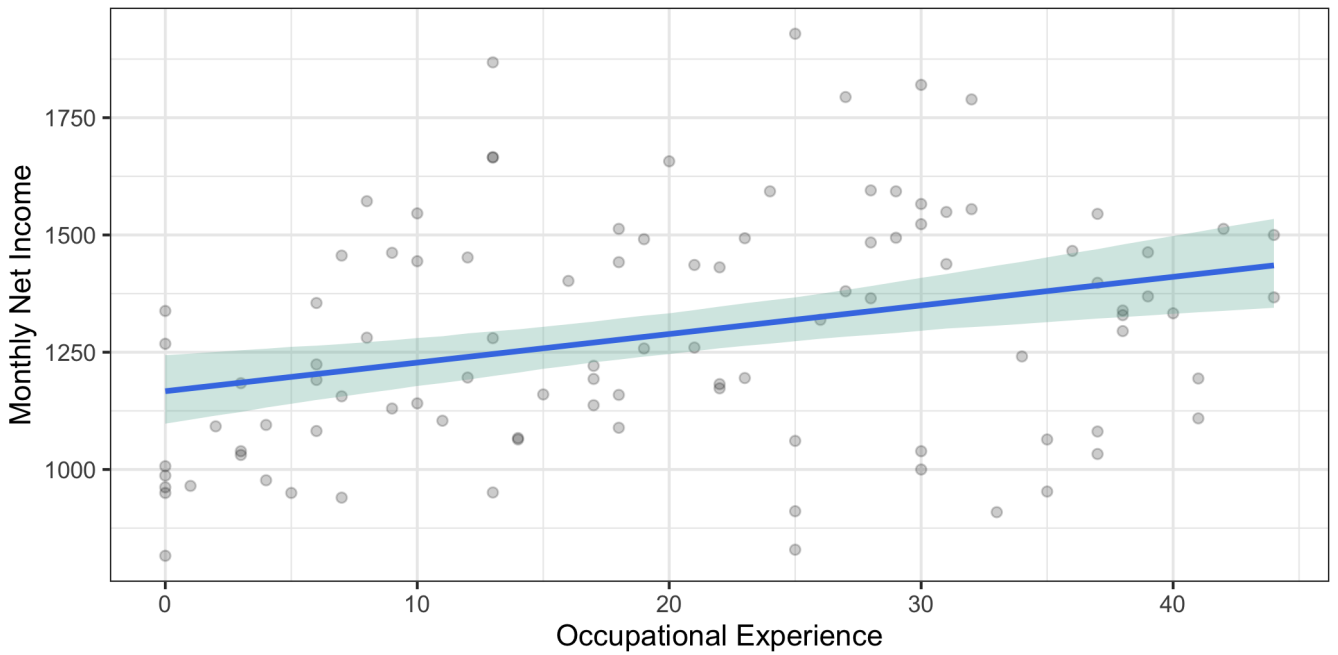
```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b.out, type = "perc", index = 1)
##
## Intervals :
## Level     Percentile
## 95%   (1314, 1452 )
## Calculations and Intervals on Original Scale
```

```r
b.ci <- t(sapply(1:nrow(incex2), function(x) boot.ci(b.out, index = x
dimnames(b.ci) <- list(rownames(incex2), c('lower', 'upper'))
kable(head(b.ci, 4))
```

lower	upper
1314.181	1452.194
1148.314	1263.936
1318.056	1461.293
1324.896	1479.626

Example: ggplot with bootstrap confidence intervals

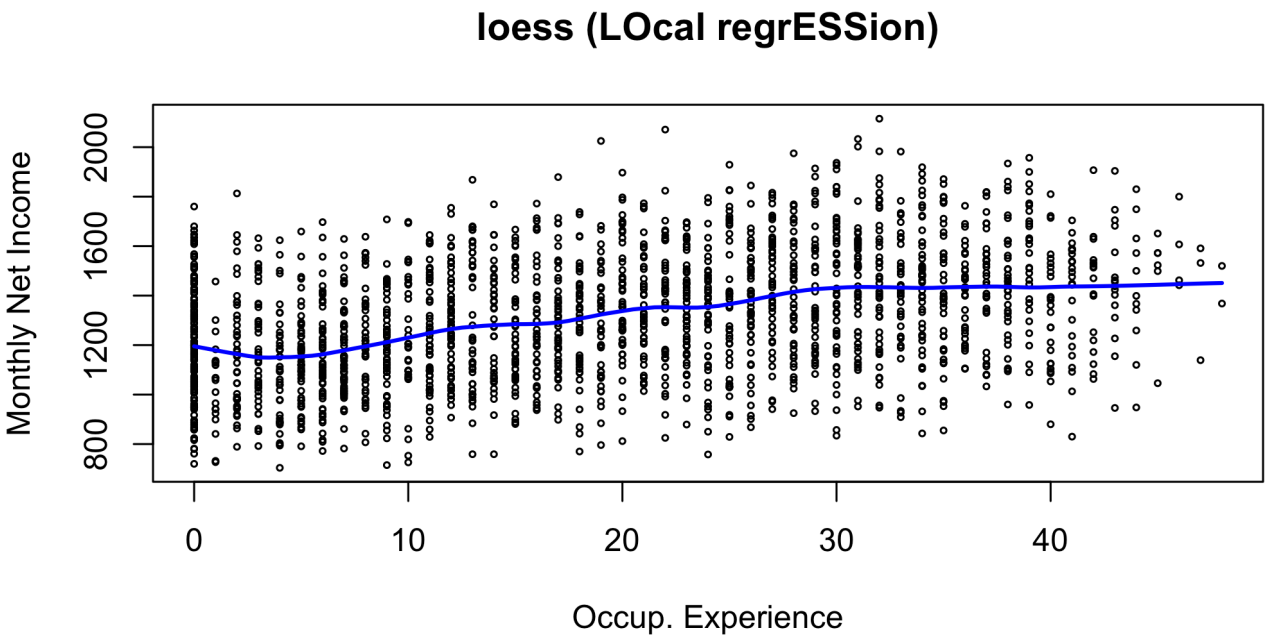
```
incex4 <- cbind(incex2, b.ci) # combine two datasets
ggplot(incex4, aes(x=oexp, y=income)) + geom_point(alpha=0.2) + labs(
        geom_smooth(method='lm', formula= y~x, se = FALSE) +
        geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.3, fi
```



Loess with confidence intervals

loess

- Scatterplot with loess regression



loess: Base R

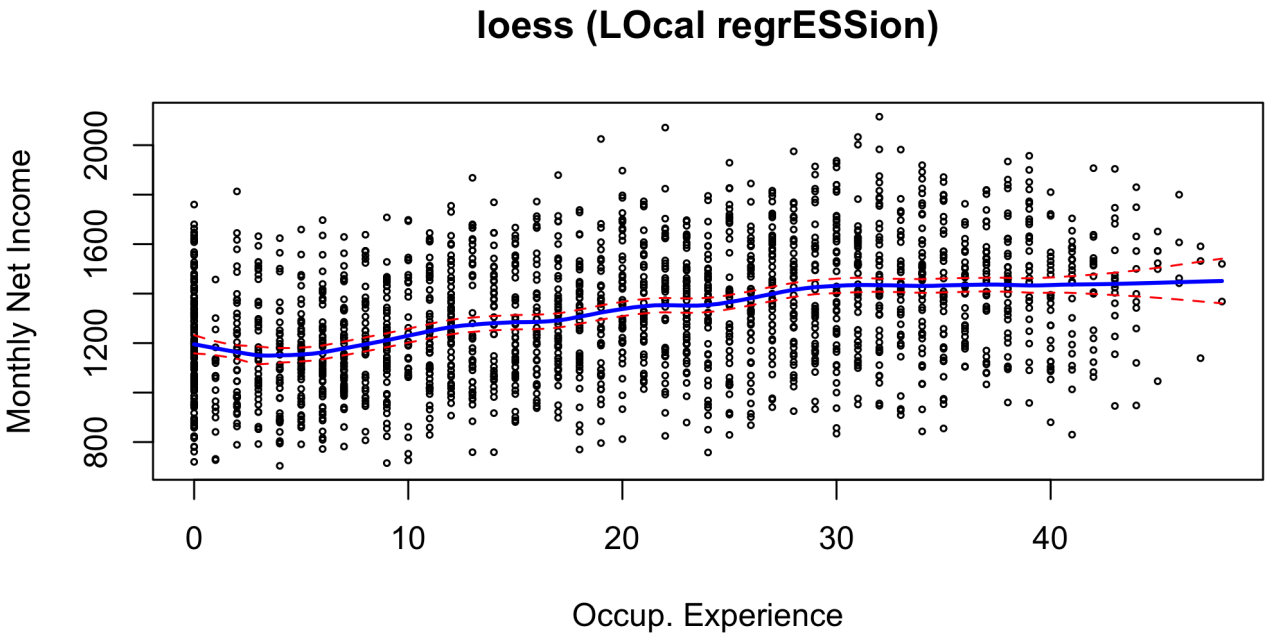
- We can add confidence intervals/envelopes for non-parametric, loess regression.
- Get standard errors for predicted means and construct confidence intervals.

```
loess.pred <- predict(out.lss, x.val, se=TRUE)
str(loess.pred)
```

```
## List of 4
##  $ fit           : num [1:97] 1195 1187 1179 1171 1164 ...
##  $ se.fit        : num [1:97] 18.8 16.2 14.4 13.7 13.9 ...
##  $ residual.scale: num 234
##  $ df            : num 1911
```

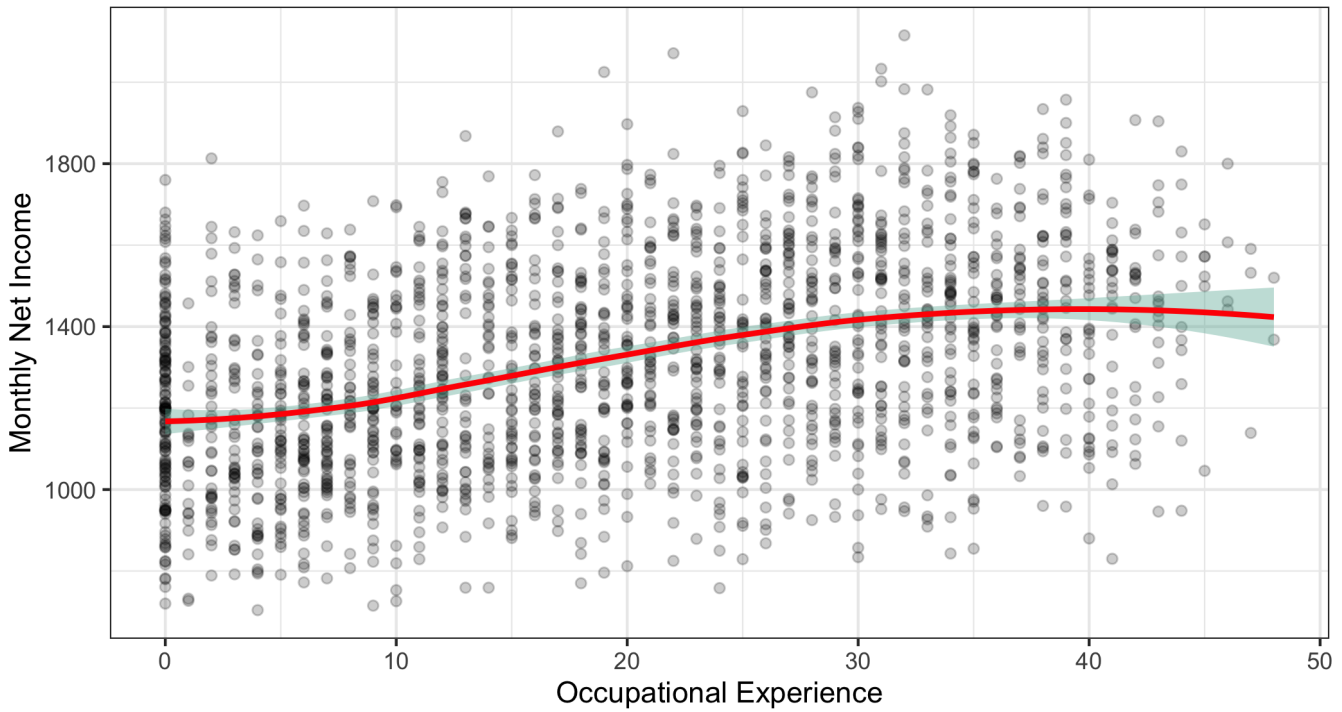
loess: Base R

```
plot(income ~ oexp, data = incex, cex = .4, xlab = 'Occup. Experience
     ylab = 'Monthly Net Income', main = 'loess (LOcal regrESSion)')
lines(x.val, y.pred, col = 'blue', lwd = 2)
lines(x.val, y.pred - qt(0.975, loess.pred$df)*loess.pred$se, lty=2,
lines(x.val, y.pred + qt(0.975, loess.pred$df)*loess.pred$se, lty=2,
```



loess: ggplot2

```
ggplot(incex, aes(x=oexp, y=income)) + geom_point(alpha=0.2) + labs(x
        geom_smooth(method='loess', formula= y~x, col="red",
                    se = TRUE, level = 0.95, fill = "#69b3a2") + ther
```



Binary outcome: logistic regression

Binary outcome

- We now want to use binary outcome instead of continuous outcome. Let's use SDS data (SCS_QE.sav).

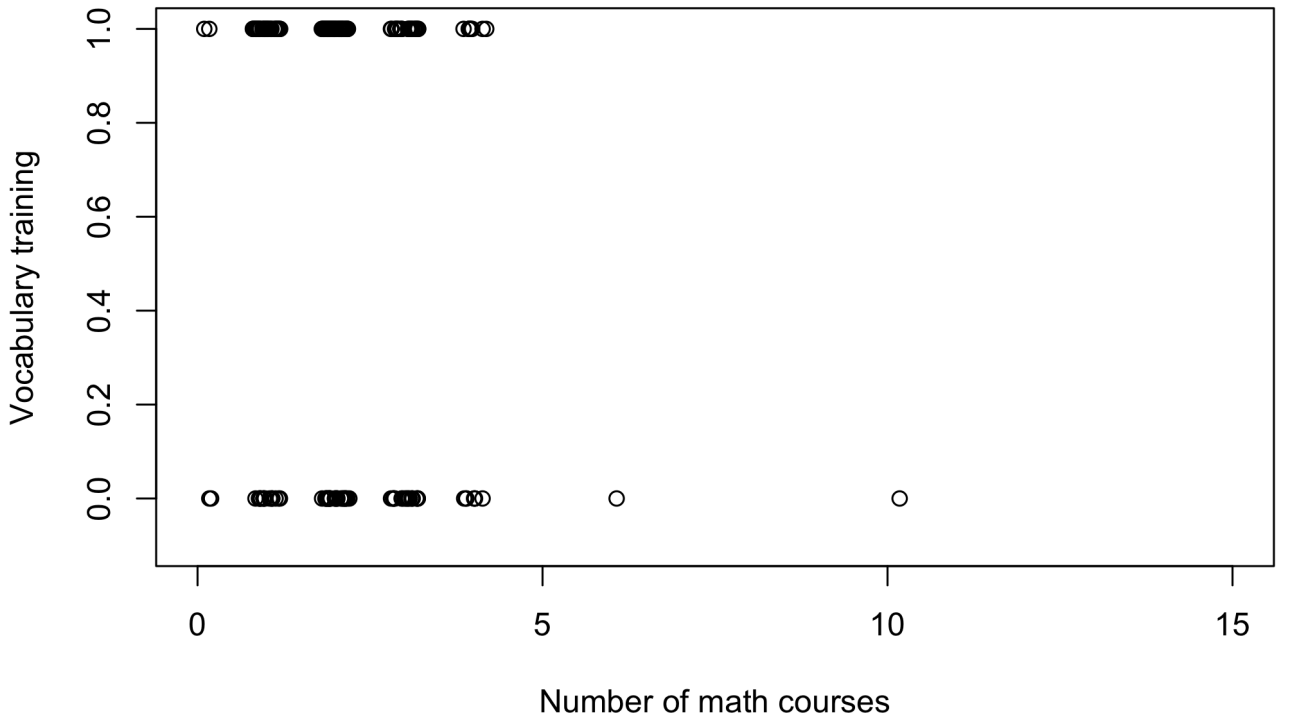
```
scs$voc <- ifelse(scs$vm == 'Vocabulary', 1, 0) # factor to numeric

kable(head(scs[, c("numbmath", "vm", "voc")]))
```

numbmath	vm	voc
2	Mathematics	0
2	Mathematics	0
1	Mathematics	0
2	Vocabulary	1
2	Vocabulary	1
2	Vocabulary	1

R Example: Base R

- You may want to jitter data points to avoid overlapping points.

```
par(mar=c(4, 4, 0.5, 2)) # bottom, left, top, right
plot(voc ~ jitter(numbmath), data = scs, xlim = c(0, 15), ylim = c(-.
   xlab = 'Number of math courses', ylab = 'Vocabulary training')
```



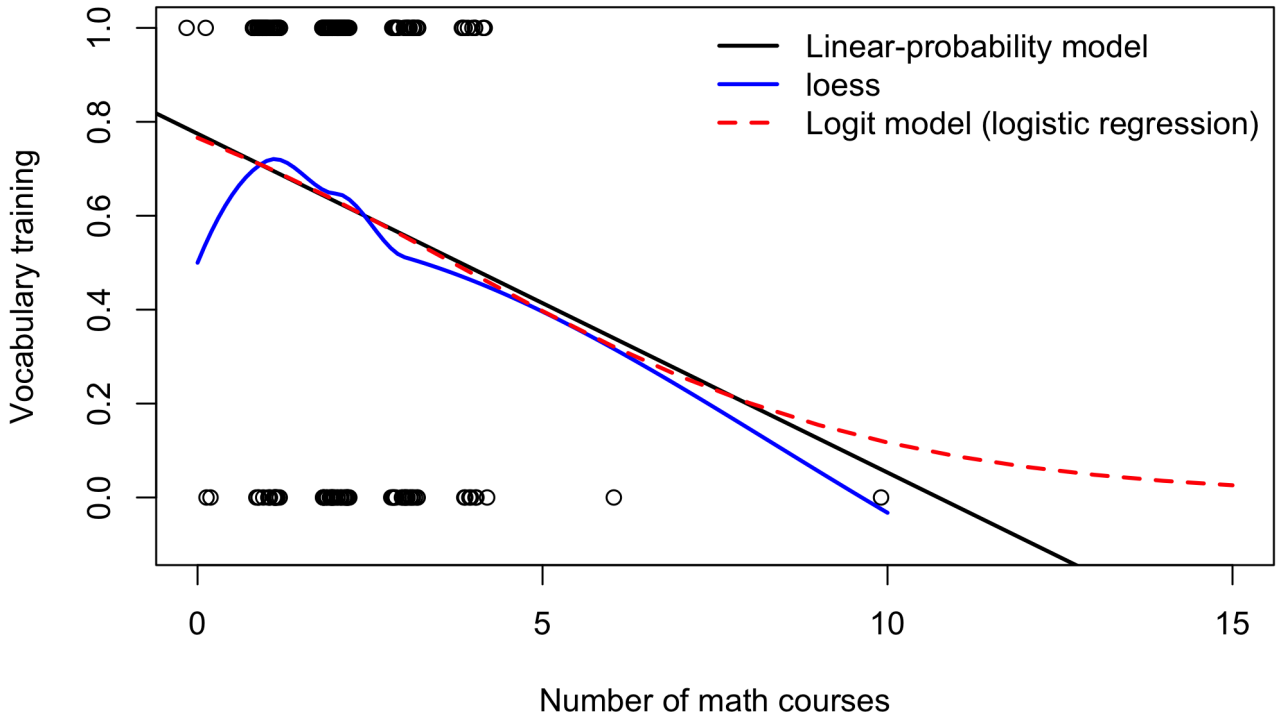
R Example: Base R

- Add regression lines.

```
par(mar=c(4, 4, 0.5, 2)) # bottom, left, top, right

plot(voc ~ jitter(numbmath), data = scs, xlim = c(0, 15), ylim = c(-.
   xlab = 'Number of math courses', ylab = 'Vocabulary training')

abline(out.lm, lwd = 2) # add linear reg
lines(seq(0, 10, .1), predict(out.smth, data.frame(numbmath = seq(0,
lines(0:15, predict(out.glm, data.frame(numbmath = 0:15), type = 'res
  lwd = 2, col = 'red', lty = 2) # add logistic reg.
legend('topright', c('Linear-probability model', 'loess', 'Logit mode
   col = c('black', 'blue', 'red'), lty = c(1, 1, 2), lwd = 2, bty =
```

R Example: Base R

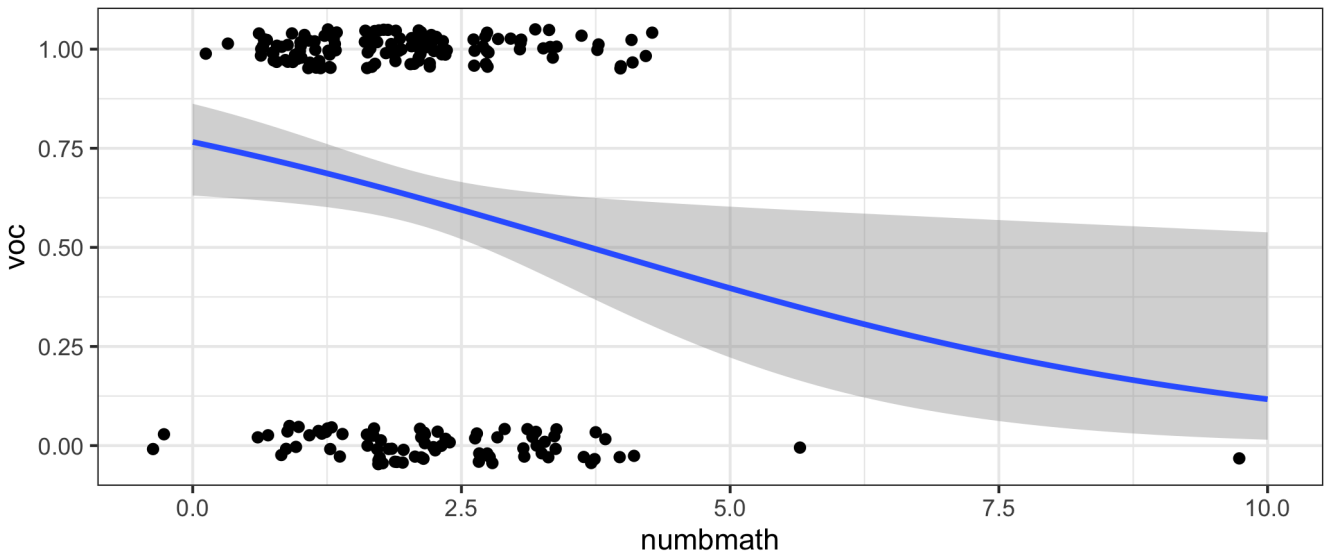
- Add regression lines.



Example: ggplot2

- To fit logistic regression in package `ggplot2`, you need to use a numeric outcome vector lying between 0 and 1.
- Try with factor `vm` instead of numeric `voc`.

```
ggplot(scs, aes(x=numbmath, y=voc)) +
  geom_jitter(height = 0.05) +
  geom_smooth(method = "glm", method.args = list(family = "binomial"))
```



Example: ggplot2

- You can fit a more flexible model. You can exercise more control and see whether it's a good model or not.

```
ggplot(scs, aes(x=numbmath, y=voc)) +
  geom_jitter(height = 0.05) +
  geom_smooth(method = "glm", formula = y ~ splines::ns(x, 2), methoc
```