# Statistic Review With R

Ryan Zhang

October 12, 2015

# Population V.S. Sample

- Population: all the individuals we are interested in
  - the characteristic of the population is called a parameter
  - example: population mean $\mu_X$
- Sample: some of the individuals draw from the population
  - the characteristic of the sample is called a statistic
  - example: sample mean $\hat{\mu}_X$
- Statistical Inference:
  - from sample statistic to population parameter

# Population V.S. Sample

```
set.seed(123)
pop = rnorm(9999,170,4)
sample1 = sample(pop, 50)
print(paste("population mean is ", round(mean(pop),2)))
print(paste("sample mean is ", round(mean(sample1),2)))
```

```
## [1] "population mean is  169.99"
## [1] "sample mean is  169.97"
```

- Mean: Arithmic Average
  - $$\bar{x} = \frac{\Sigma_i x_i}{n}$$
  - Not robust, affected by outlier
- Median: The middle value
  - More robust
- Mode: Most frequenist outcome
  - Only meaningful measure for categorical variable

```
myMean <- function(data) return(sum(data)/length(data))
myMedian <- function(data){
    data <- sort(data)
    if (length(data)%%2==1) data[(length(data)+1)/2]
    else (data[length(data)/2]+data[length(data)/2+1])/2
    }
print(paste("sample mean is",round(myMean(sample1),2)))
print(paste("sample median is ",round(myMedian(sample1),2)))
```

```
## [1] "sample mean is 169.97"
## [1] "sample median is  169.34"
```

```r
myMode <- function(data){
    freqTable <- table(data)
    return(freqTable[freqTable == max(freqTable)])
}
print(myMode(c(1,1,1,2,2,2,3,3,4,5,6,6,7,7,7)))
```

```
## data
## 1 2 7
## 3 3 3
```

- Variance, Skewness and Kurtosis are different moments
- k-th moment:

$$\frac{\Sigma(X - \bar{X})^k}{N}$$

- Variance: k = 2
  - Standard Deviation is Variance adjusted via SQRT to get back to the original unit of measure
- Skewness: k = 3
  - $Skewness < 0 \implies$ negative skew
  - $Skewness > 0 \implies$ positive skew
- Kurtosis: k = 4
  - The kurtosis of a normal distribution $N(\mu, \sigma^2)$ is $3\sigma^4$
  - Higher Kurtosis means fatter tails.

- Frequency: the number of times a certain outcome occurs
- If we know the frequencies of all possible outcomes, we can calculate the probability of each single one of them:

$$Pr(outcome_j) = \frac{\text{freq of outcome j}}{\Sigma_i \text{freq of outcome i}} \times 100$$

- For discrete variable, the frequency is literally what it is
- For continuous variable, we need to create intervals and assign values into intervals in order to get the frequency.

- What is $Pr(165 \leq X < 167.5)$ in our sample1?

```
freq_j = sum((sample1 >= 165) & (sample1 < 167.5))
freq_total = length(sample1)
round(freq_j/freq_total,2)*100
```

```
## [1] 22
```

- This 22% is our emperical probability.

- What is $Pr(X < 170)$ in our sample1?

```
freq_j = sum(sample1 < 170)
freq_total = length(sample1)
round(freq_j/freq_total,2)*100
```

```
## [1] 54
```

- This 54% is our emperical probability.

- What is $Pr(X > 175)$ in our sample1?

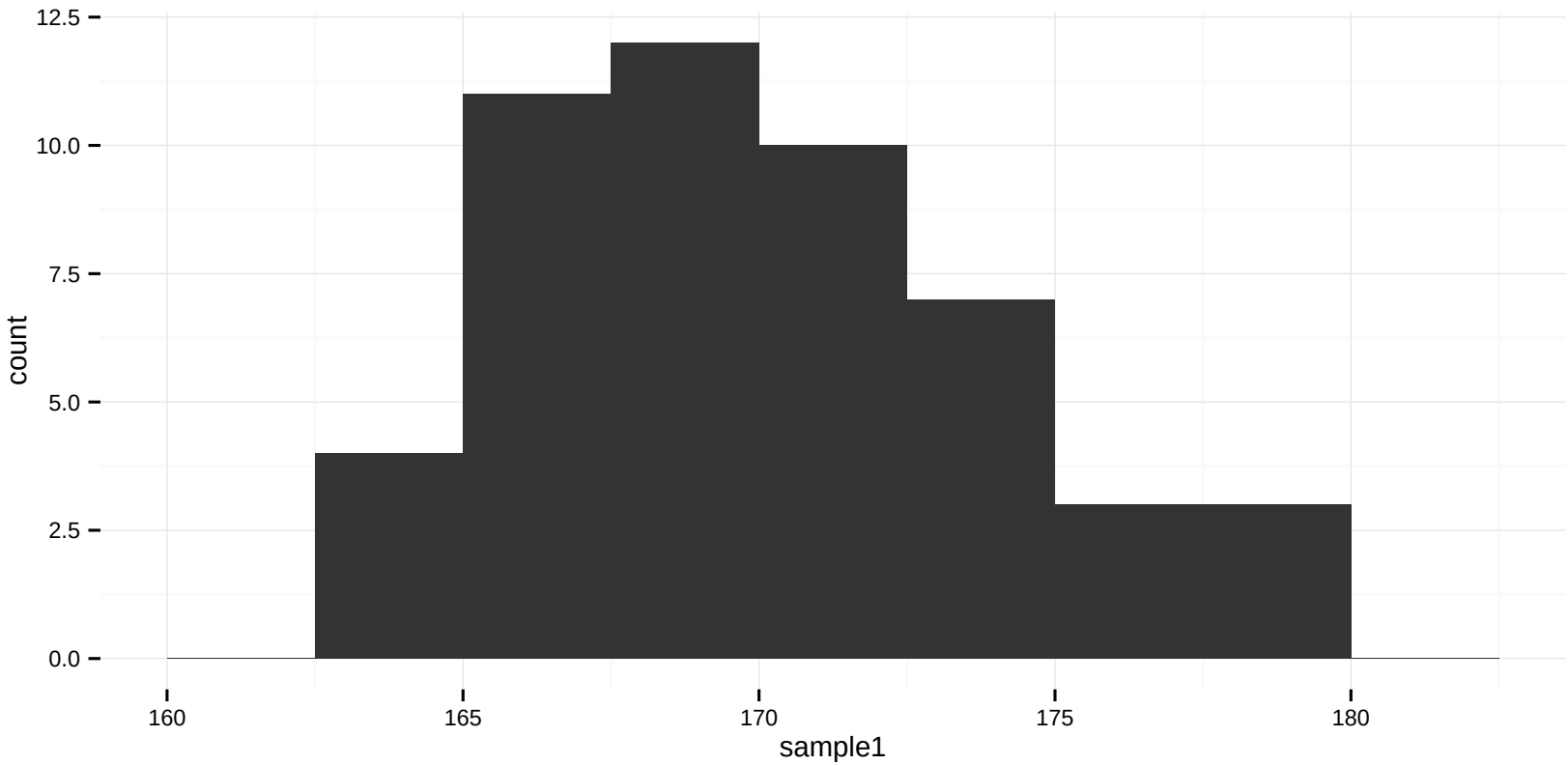
```
freq_j = sum(sample1 > 175)
freq_total = length(sample1)
round(freq_j/freq_total,2)*100
```

```
## [1] 12
```

- This 12% is our emperical probability.

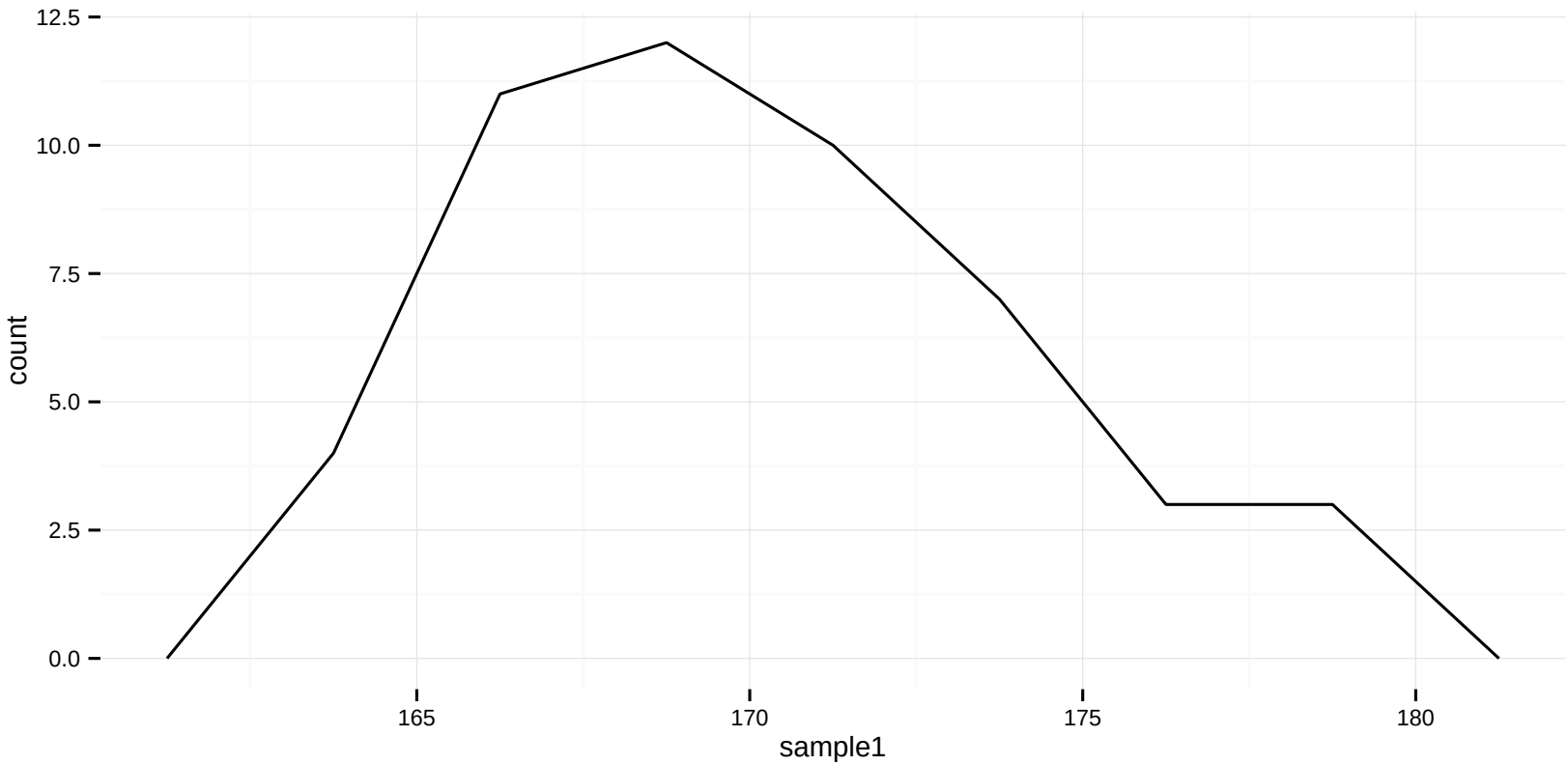
# Histogram

```
qplot(x = sample1, binwidth = 2.5)
```



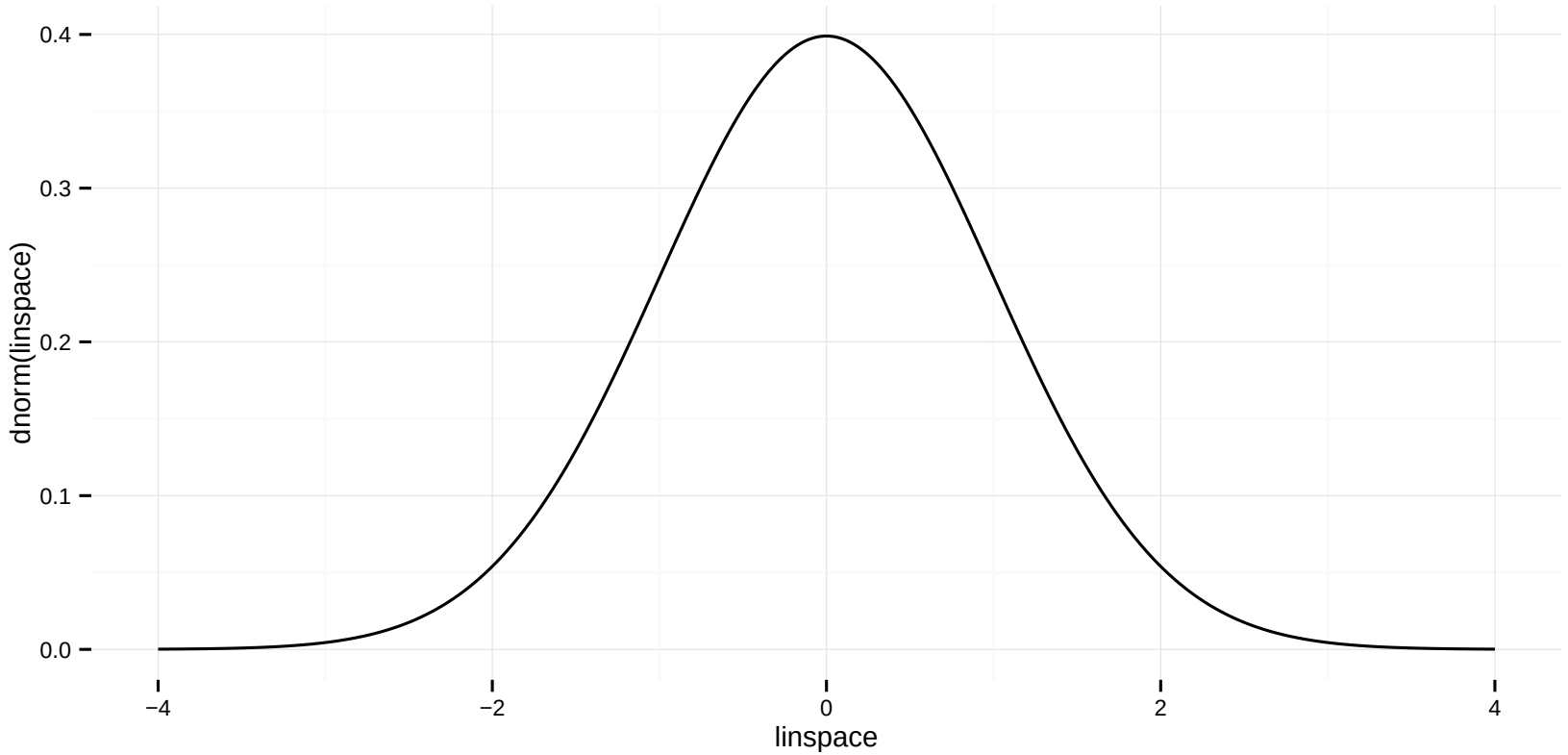
# Freqency Polygon

```
qplot(x = sample1, geom = "freqpoly", binwidth = 2.5)
```



# Normal Distribution

```
linspace = seq(-4,4,0.01)
qplot(x = linspace, y = dnorm(linspace), geom = "line")
```



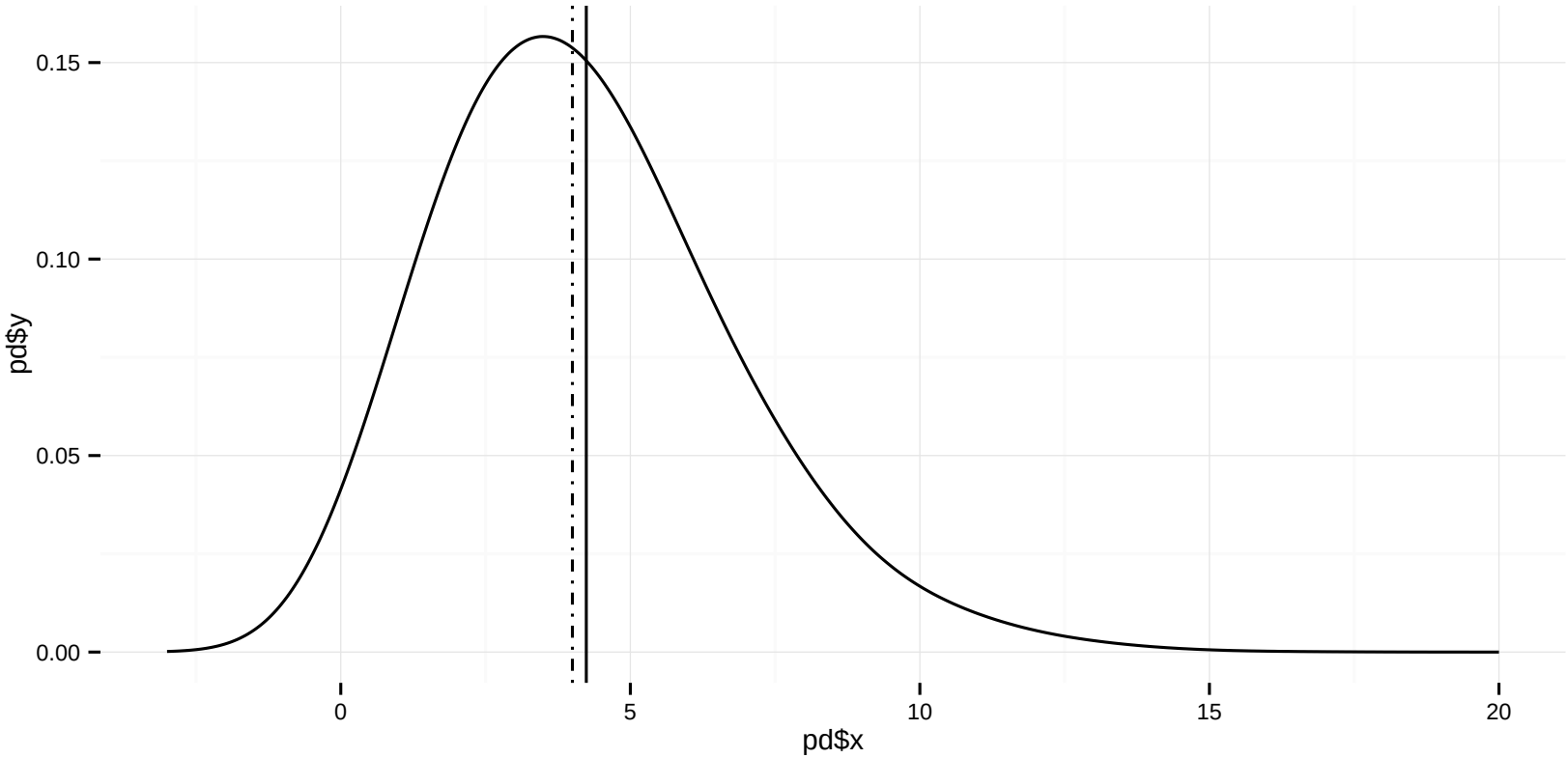
# Skewed Distribution

- Right == Positive == long tail on right hand side
- Left == Negative == long tail on left hand side

```
# some example data
set.seed(123)
binomSample <- rnbinom(9999, 10, .7)
pd <- density(binomSample, bw = 1)
```

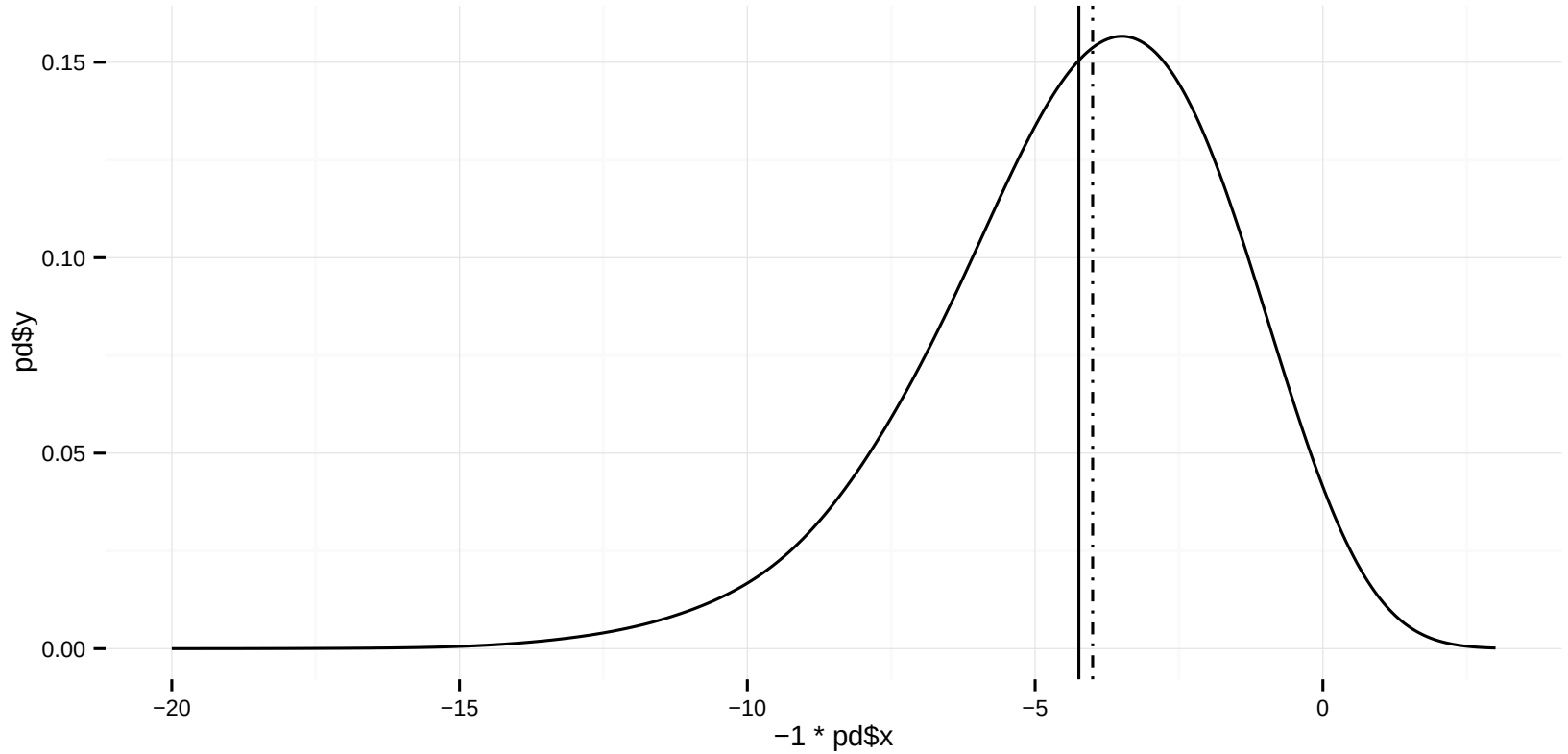
# Positive/Right Skewed Distribution

```
qplot(x = pd$x, y = pd$y, geom = "line") +
    geom_vline(xintercept = mean(binomSample)) +
    geom_vline(xintercept  = median(binomSample), linetype = 4)
```



# Negative/Left Skewed Distribution

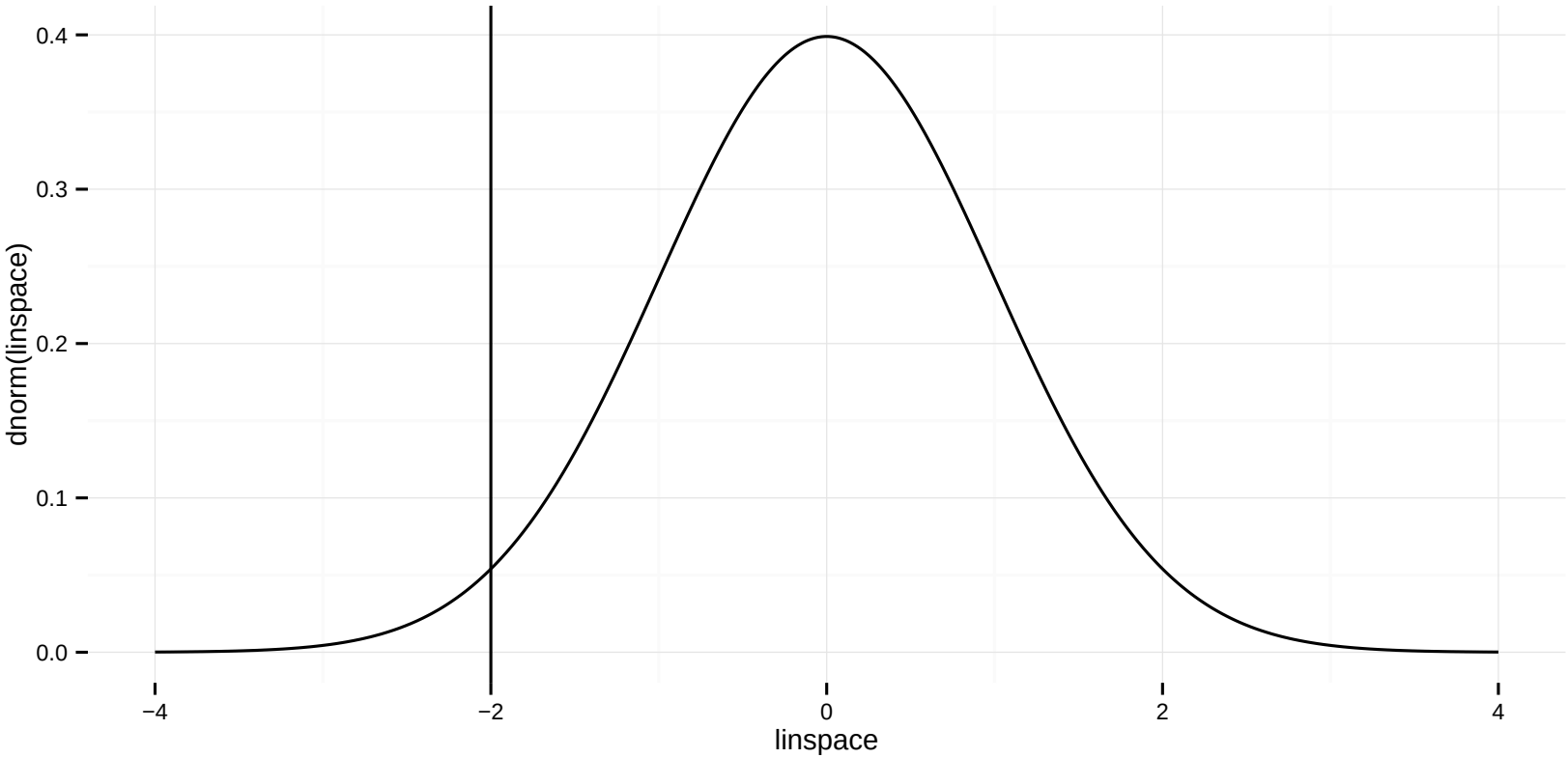
```
qplot(x = -1*pd$x, y = pd$y, geom = "line") +
    geom_vline(xintercept = -1*mean(binomSample)) +
    geom_vline(xintercept  = -1*median(binomSample), linetype = 4)
```



- If $X \sim N(0, 1)$
- What is $Pr(X \leq -2)$

```
pnorm(-2, mean = 0, sd = 1)
```

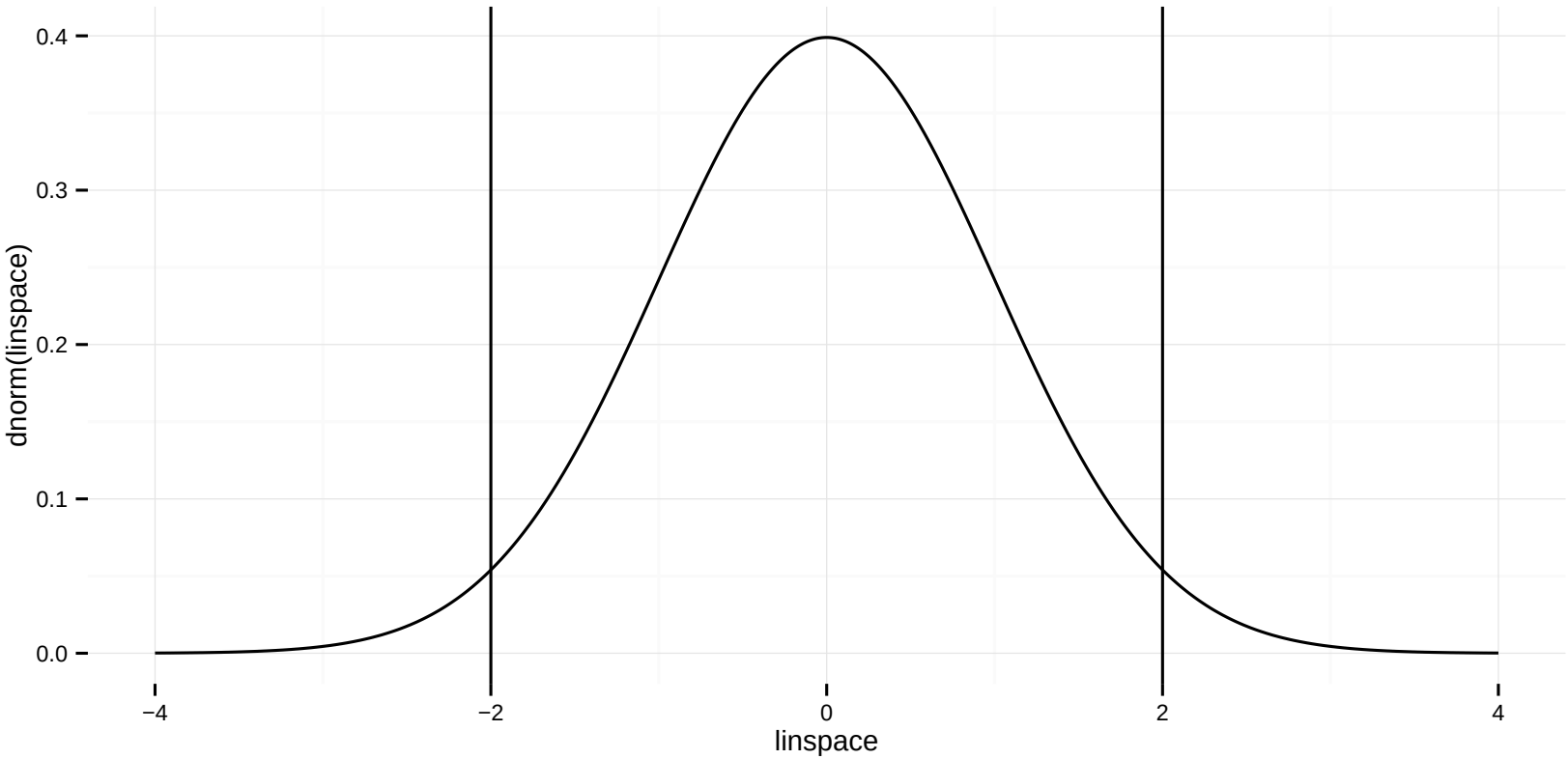
```
## [1] 0.02275013
```



- If $X \sim N(0, 1)$
- What is $Pr(-2 \leq X \leq 2)$

```
pnorm(2, mean = 0, sd = 1) - pnorm(-2, mean = 0, sd = 1)
```

```
## [1] 0.9544997
```

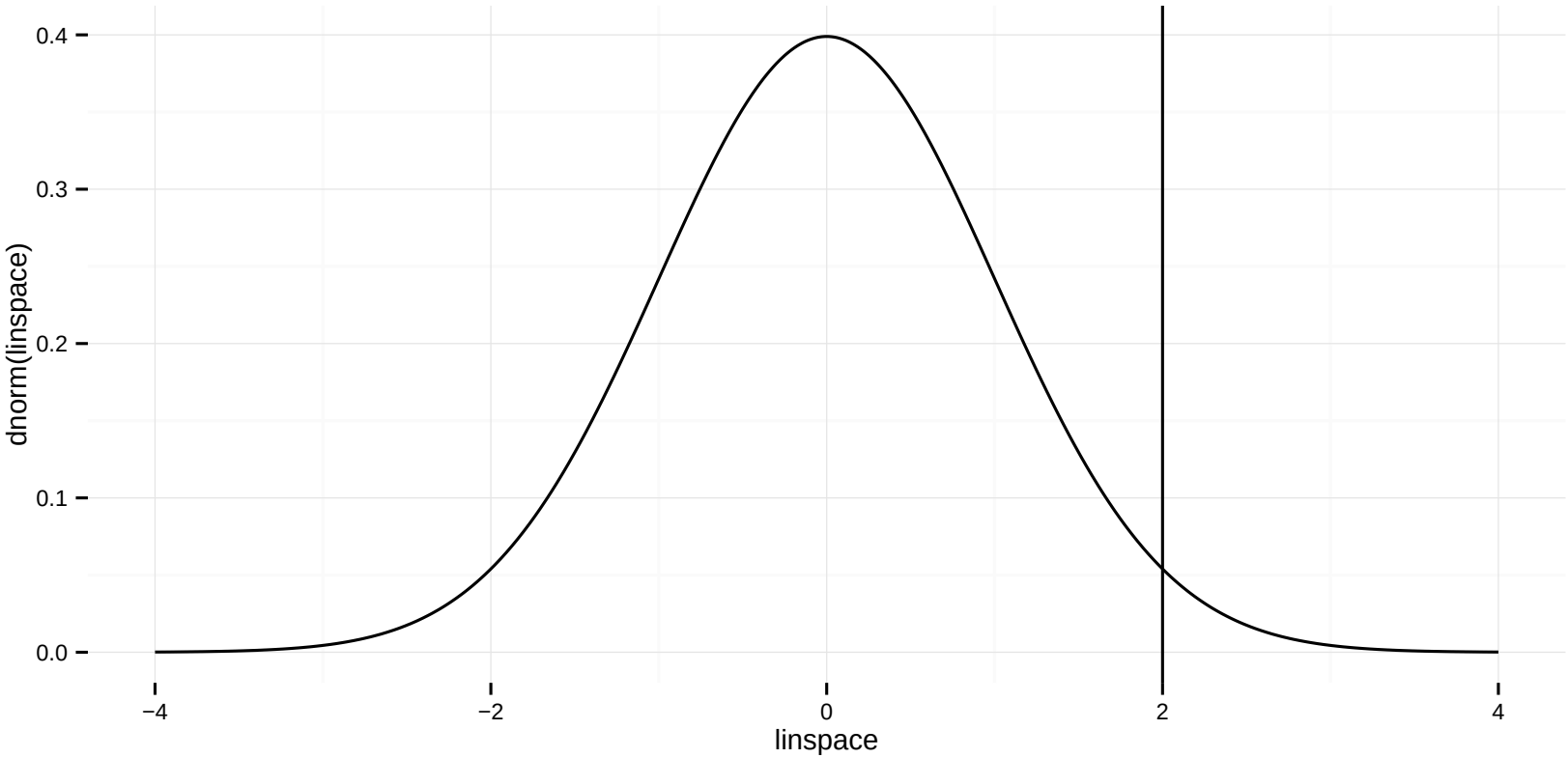


# Probability From Distributions

- If $X \sim N(0, 1)$
- What is $Pr(X \geq 2)$

```r
pnorm(2, mean = 0, sd = 1, lower.tail = F)
```

```
## [1] 0.02275013
```



# Quantile

- An lower alpha quantile is a value $q_\alpha$ such that $100 \times alpha$ percent of the data is less than or equal to it.
- In other word: $Pr(X \leq q_\alpha) = \alpha$
- Median is just the 50% quantile

```
quantile(sample1)
median(sample1)
```

```
##       0%      25%      50%      75%     100%
## 163.1585 166.8809 169.3390 172.6214 179.5953
## [1] 169.339
```

- These are our emperical quantiles

# Quantile

- The upper 0.05 quantile is often seen

```
qnorm(0.05,lower.tail = F)
```

```
## [1] 1.644854
```

- Which simple means that:
  - If $X \sim N(0, 1)$
  - Then $Pr(X \geq \text{upper } q_\alpha) = 0.05$
  - Where upper $q_\alpha = 1.644854$

# Conditional Probability

- The probability of an event happen given another event happened
- Denoted as:
  - $Pr(A|B)$
  - $Pr(A|X = x)$
  - etc.

$$Pr(Speal.Length >= 4.9)$$

$$\neq$$

$$Pr(Speal.Length >= 4.9 | Species = "setosa")$$

```
data(iris)
100*round(sum(iris$Sepal.Length >= 4.9)/nrow(iris),2)
100*round(sum(iris$Sepal.Length[iris$Species == "setosa"]
              >= 4.9)/sum(iris$Species == "setosa"),2)
```

```
## [1] 89
## [1] 68
```

$$Pr(X \geq 2 | X \sim N(0,1))$$
$$\neq$$
$$Pr(X \geq 2 | X \sim t(\nu = 30))$$

```
pnorm(2, mean = 0, sd = 1, lower.tail = F)
pt(2,df = 30, lower.tail = F)
```

```
## [1] 0.02275013
## [1] 0.02731252
```

- T-distribution has fatter tail than normal distribution
- Try it out by calculating the Kurtosis if you want
- As $\nu \to \infty$ T distribution will be more close to Standard Normal distribution

# Conditional Probability View of Quantile

```
qnorm(0.05,mean = 0, sd = 1, lower.tail = F)
```

```
## [1] 1.644854
```

- If $X \sim N(0,1)$
- Then $Pr(X \geq 1.644854) = 0.05$
- Expressed as conditional probability:
  - $Pr(X \geq 1.644854 | X \sim N(0,1)) = 0.05$

- sample mean depend on the sample

```
mean(sample1)
```

```
## [1] 169.9741
```

- Is $\hat{\mu}_{X|sample1}$

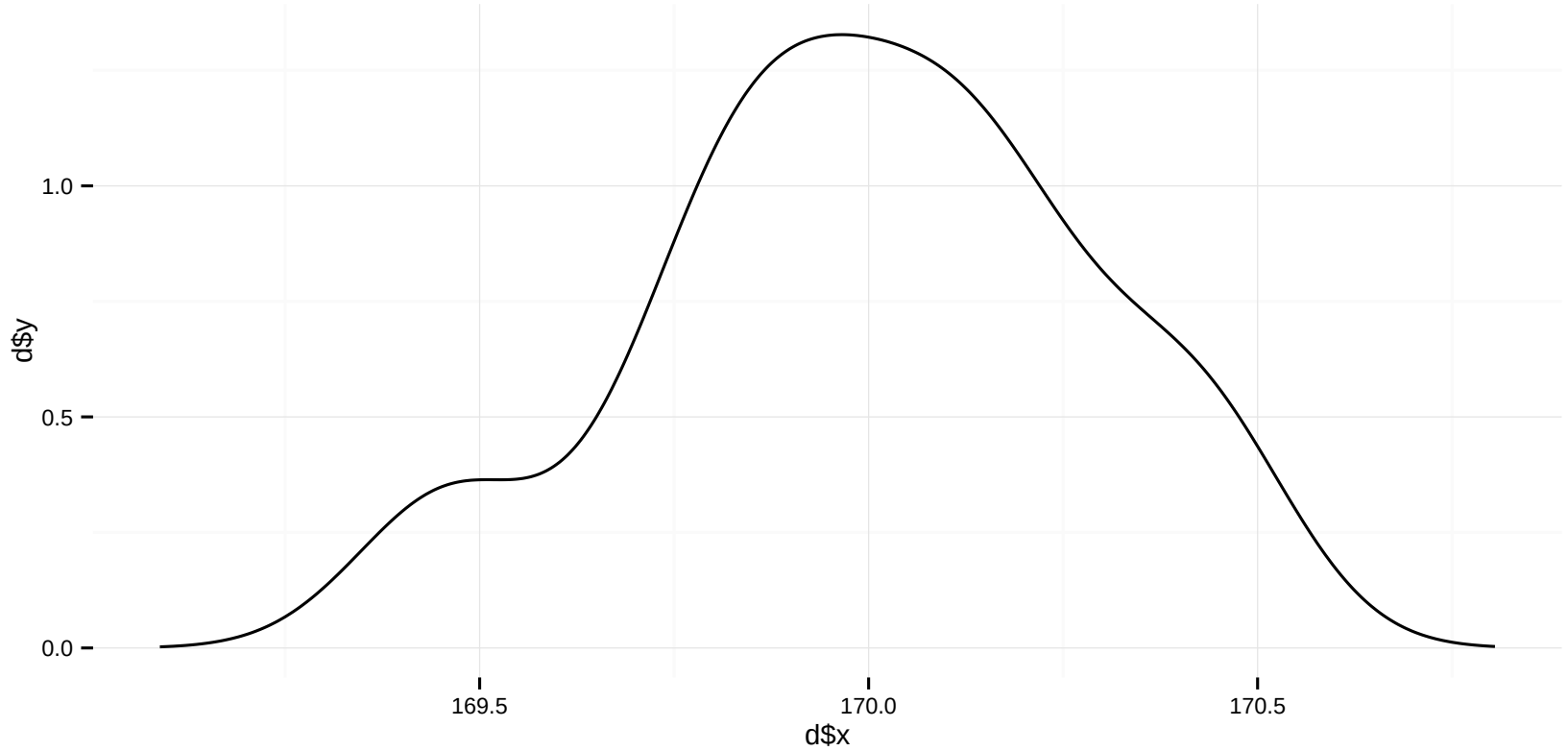
- Repeated draw samples from population
- Calculate mean on each sample

```
sample_means <- vector()
set.seed(0306)
for(i in 1:30){
    s = sample(pop,size = 200, replace = F)
    sample_means <- c(sample_means, mean(s))
}
sample_means[1:5]
```

```
## [1] 169.9021 169.8980 169.9382 169.9669 170.1715
```

- Distribution of sample means

```
d = density(sample_means)
qplot(d$x, d$y, geom = "line")
```



# Central Limit Theorem for Sample Mean

- From any population distribution, as long as the variance of that population is finite
- We draw n samples from the distribution, and calculate n sample means
- If n is sufficiently large (usually 30 will do)
- Then the distribution of the n sample means will be approximately normal

# Law of Large Numbers

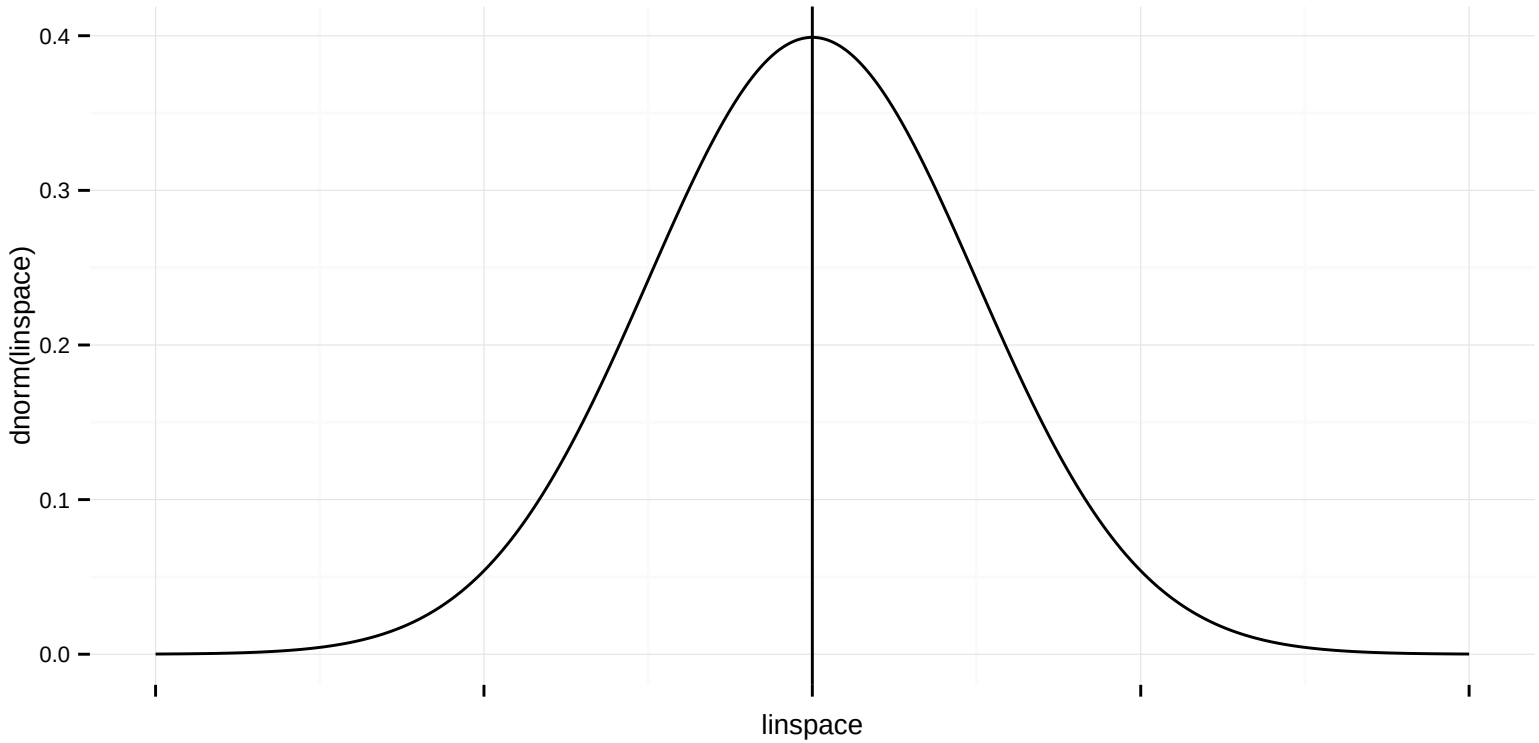
- From any population distribution, as long as the variance of that population is finite
- We draw n samples from the distribution, and calculate n sample means
- The mean of n sample means converge in probability to population mean as $n \to \infty$.

# Good News and Bad News

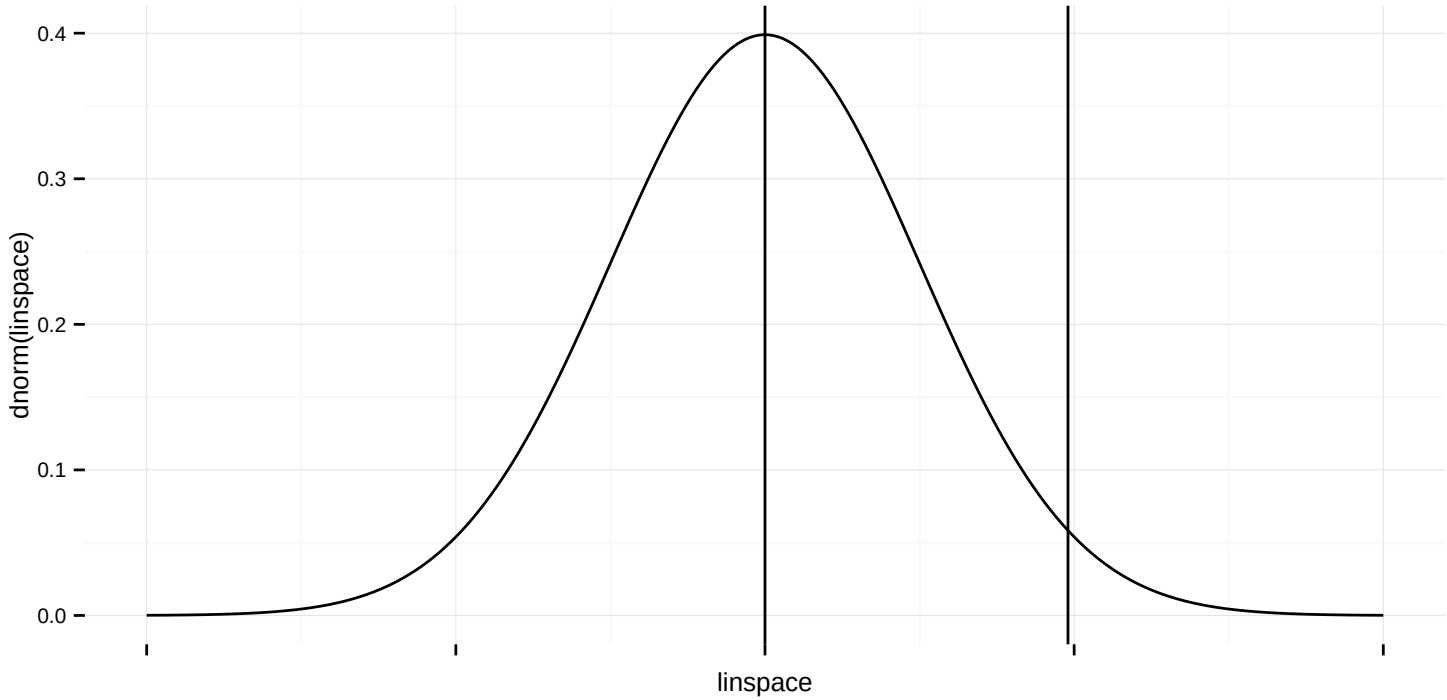
- Good News: If we can draw large numbers of samples from population
  1. The distribution of sample means will be normal
  2. The mean of sample means will be close to population mean
- Bad News: Often time all we have is a sample
- What we do next?

# Null Hypothesis Significance Testing, NHST

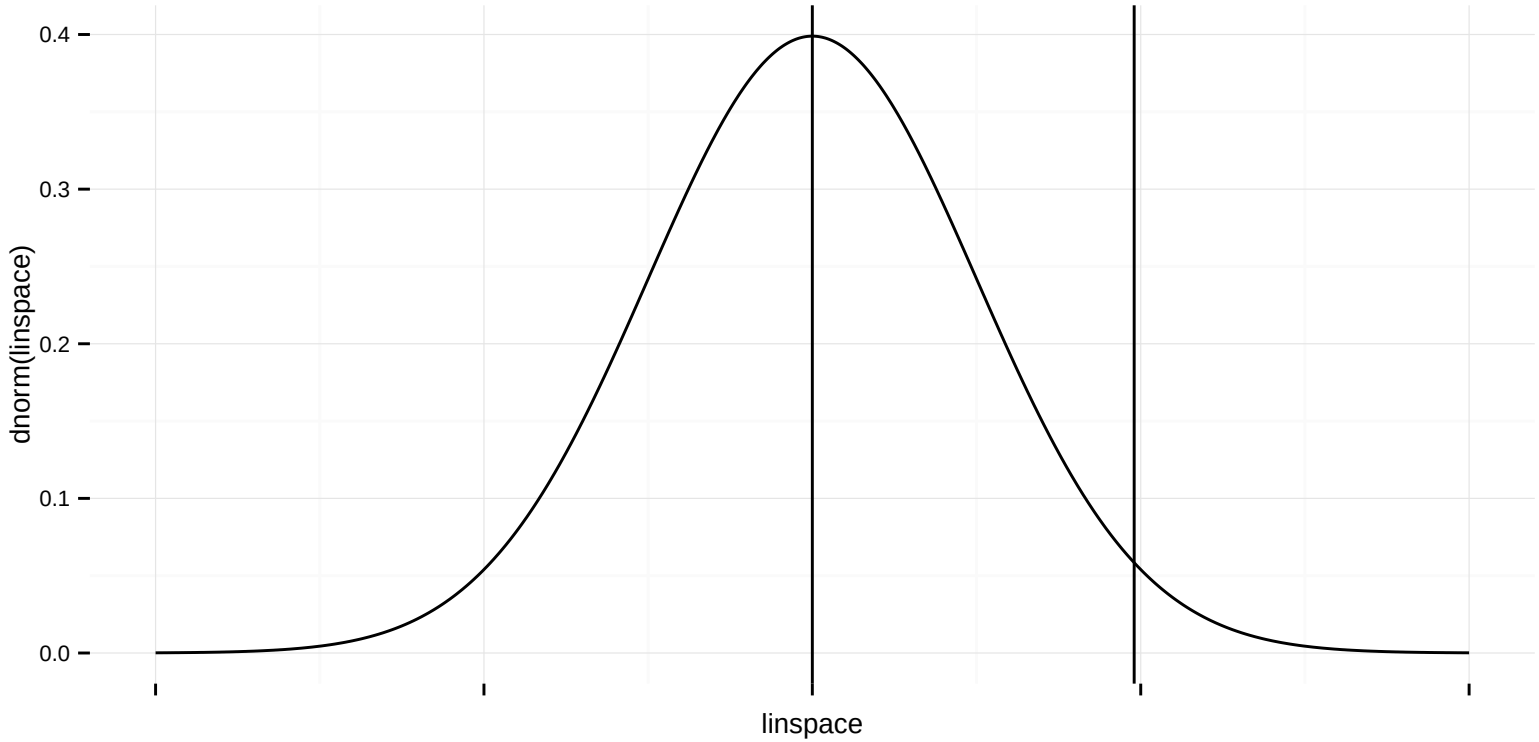
- Null hypothesis: the population mean is $\mu_0$
- If null is true, then the mean of sample means will be $\mu_0$, plus sample means normally distributed
- Esitmate standard error (standard deviation of the sampling distribution) using $\frac{\sigma}{\sqrt{n}}$
- Then the sampling distribution follows $N(\mu_0, \frac{\sigma^2}{n})$



- Given null is true, we can infer that the sampling distribution as shown in previous slide
- Then, we can ask the question:
  - What is the probability we obtain a sample mean large than or equal to the one $\hat{\mu}_X$ we have?
  - $Pr(X \geq \hat{\mu}_X | X \sim N(\mu_0, \frac{\sigma^2}{n}))$
  - $Pr(X \geq \hat{\mu}_X |$null is true$)$



- $Pr(X \geq \hat{\mu}_X | \text{null is true})$ the p-value is a upper quantile
- Meaning, what percentage of sample means have a value greater than or equal to the one we had.
- If the percentage/ probability is very small, we tend to believe that the null is not true.
- Then we rejcet the null.
- Remark: This is stupid logic.



# Procedure of NHST of Sample Mean

1. Specify null and alternative hypothesis
2. Calculate test statistic
   - $z = \frac{\hat{\mu}_X - \mu_0}{\frac{\sigma}{\sqrt{n}}}$ if population variance is known (do we really have this case?)
   - $t = \frac{\hat{\mu}_X - \mu_0}{\frac{\hat{\sigma}}{\sqrt{n}}}$ if population variance is unknown
3. Calculate the p value
   - $Pr(X \geq \hat{\mu}_X | \text{null is true})$ suppose we we do a right-tailed test
   - Which is equal to $Pr(Z \geq z | Z \sim N(0,1))$ if population variance is known
   - and $Pr(T \geq t | T \sim t(\nu))$ if population variance is unknown
4. Make judgement

# Various form of NHST of the mean

| $H_0$ | $H_A$ | p value |
|---|---|---|
| $\mu = \mu_0$ | $\mu \neq \mu_0$ | $2Pr(Z \geq \|z\|)$ |
| $\mu = \mu_0$ | $\mu \geq \mu_0$ | $Pr(Z \geq z)$ |
| $\mu = \mu_0$ | $\mu \leq \mu_0$ | $Pr(Z \leq z)$ |

```
t.test(sample1,mu = 172, alternative = "two.sided")
t.test(sample1,mu = 172, alternative = "greater")
t.test(sample1,mu = 172, alternative = "less")
```

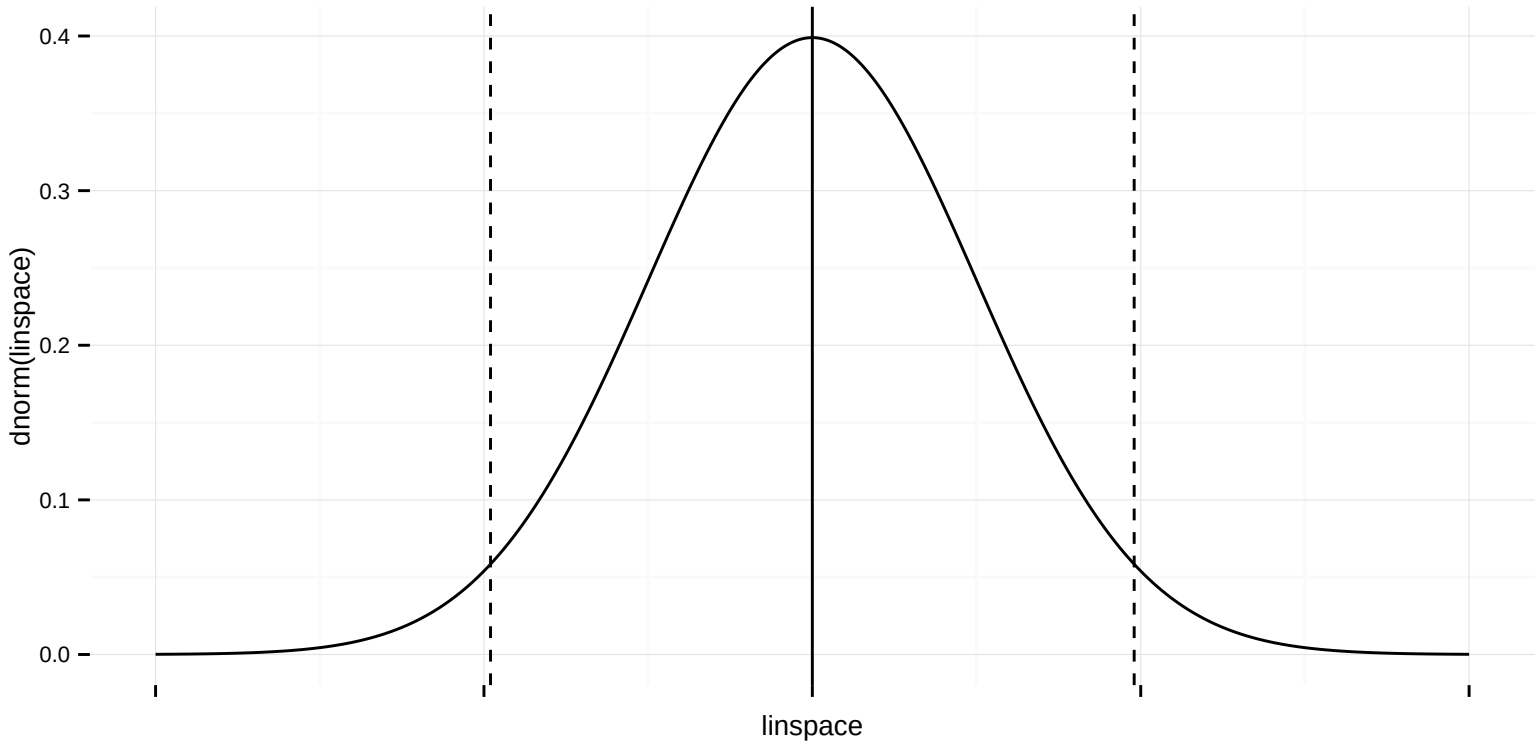
# NHST One sample T.Test in R

```r
t.test(sample1,mu = 172, alternative = "less")
```

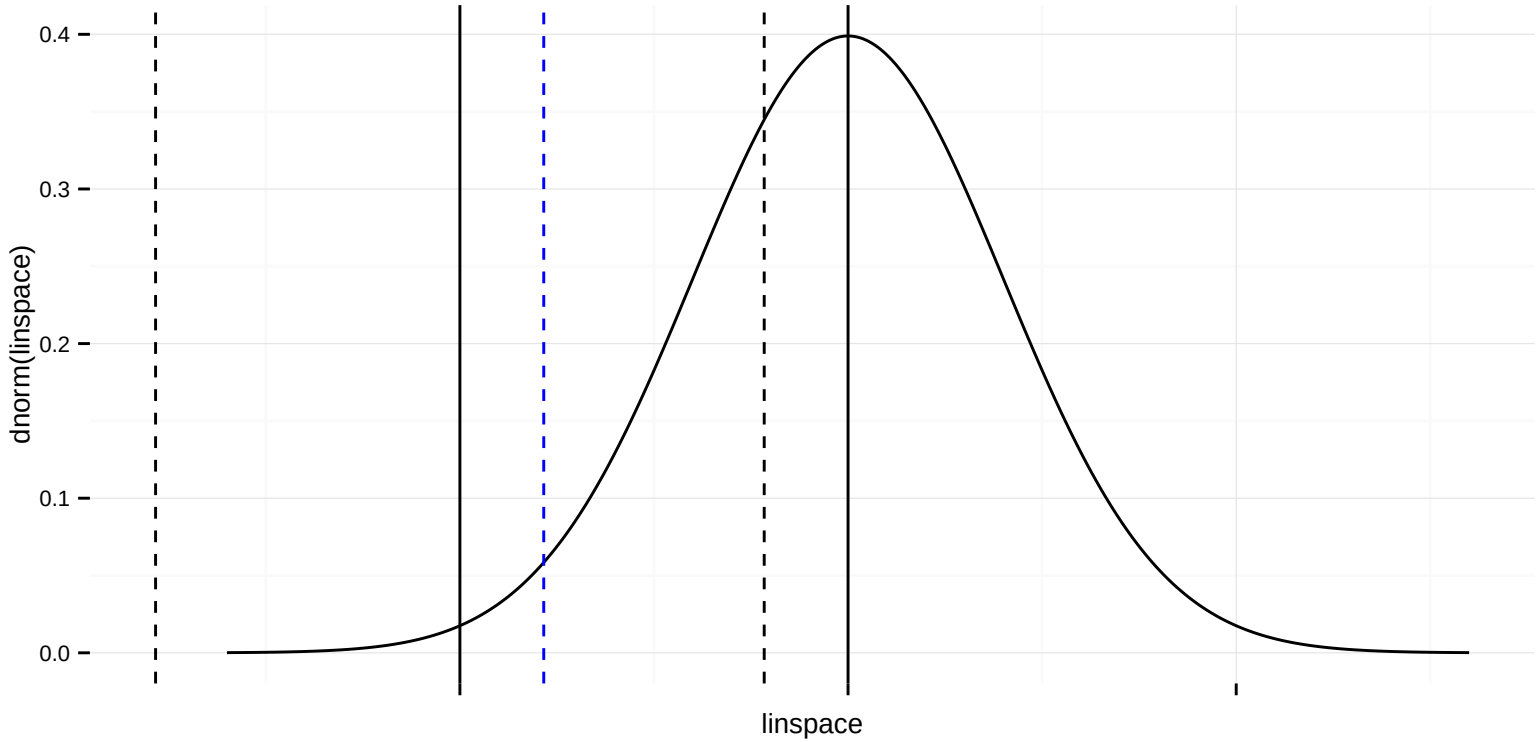
```
##
##  One Sample t-test
##
## data:  sample1
## t = -3.5813, df = 49, p-value = 0.000392
## alternative hypothesis: true mean is less than 172
## 95 percent confidence interval:
##      -Inf 170.9225
## sample estimates:
## mean of x
##  169.9741
```

# Confidence Interval

- If we draw a 95% interval around the mean of sampling distribution
- Ofcourse we will include the true mean of sampling distribution

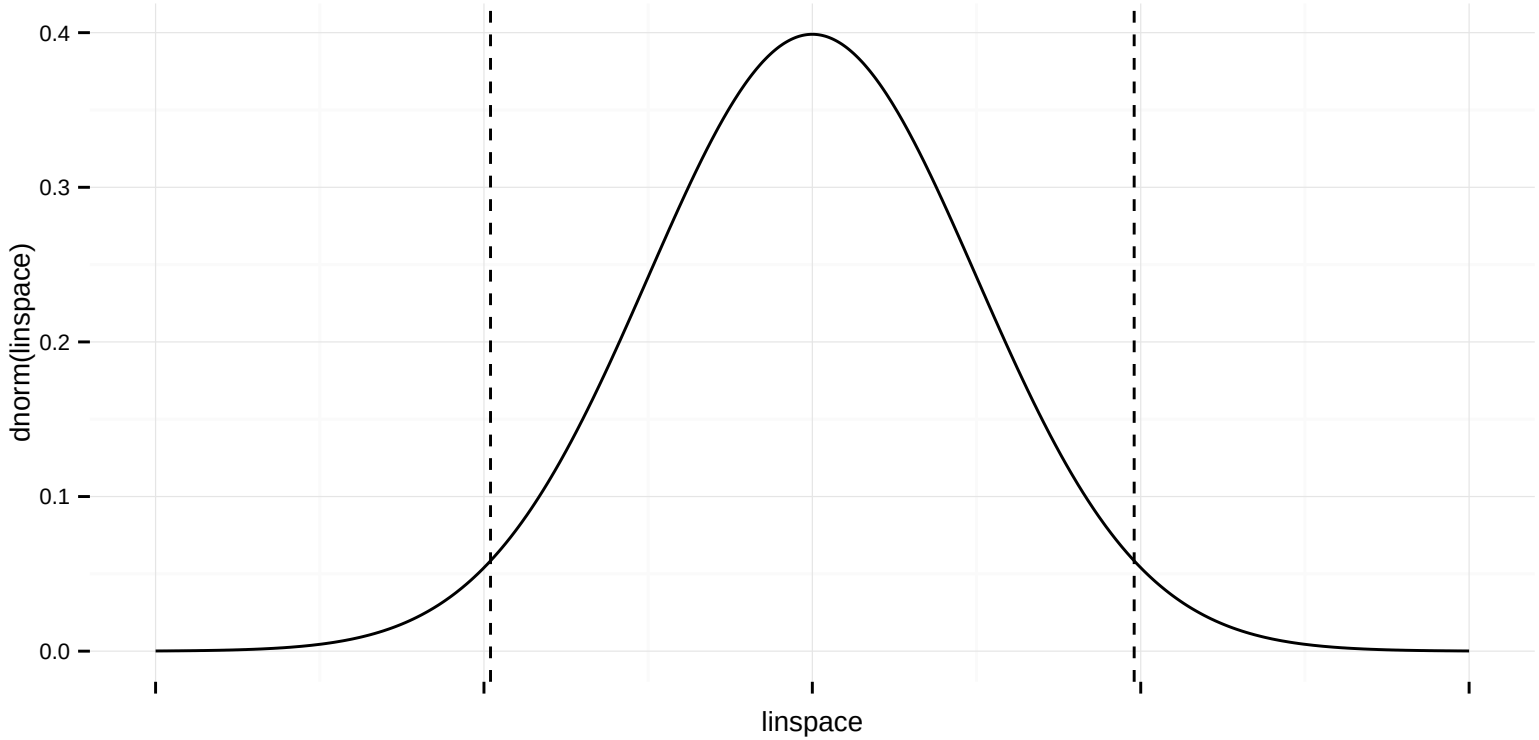


- If we draw a 95% interval around a value that is smaller than the lower 0.025 quantile
- The mean of sampling distribution will not be included



- The formula is $\hat{\mu}_X \pm t_{\frac{\alpha}{2},\nu} se$
- Back to standard scale it is simply $0 \pm t_{\frac{\alpha}{2},\nu}$
- In conditional probabilty sense:

$$Pr(-t_{\frac{\alpha}{2},\nu} \leq T \leq t_{\frac{\alpha}{2},\nu} | T \sim t(\nu)) = 0.95$$



- How often we get a sample mean below the lower 0.025 quantile or larger than the upper 0.025 quanitle?
- $0.025 + 0.025 = 0.05 = 5\%$
- That is to say 5% of the intervals we constructed this way will not include the true mean of sampling distribution, which according to Law of Large Numbers should be the true population mean.

- If a null hypothesis $\mu = \mu_0$ can be rejected at a $\alpha$ significance level
- Then the $100(1-\alpha)$ percent confidence interval will not contain $\mu_0$

```
t.test(sample1,mu = 172, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  sample1
## t = -3.5813, df = 49, p-value = 0.0007839
## alternative hypothesis: true mean is not equal to 172
## 95 percent confidence interval:
##  168.8373 171.1109
## sample estimates:
## mean of x
##  169.9741
```

# Paired Two Sample T-Test

- Same measure score before treatment and after treatment
- $H_0 : \mu_1 - \mu_2 = d_0$

```
set.seed(1106)
sample2 = sample(pop, size = 50, replace = T) + 0.8
t.test(sample1, sample2, mu = 0, paired = T)
```

```
##
##  Paired t-test
##
## data:  sample1 and sample2
## t = -2.2143, df = 49, p-value = 0.03149
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.2614750 -0.1580463
## sample estimates:
## mean of the differences
##               -1.709761
```

# Paired Two Sample T-Test

- It is equivalent to a one sample test on the differences

```
t.test(sample1-sample2)
```

```
##
##  One Sample t-test
##
## data:  sample1 - sample2
## t = -2.2143, df = 49, p-value = 0.03149
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -3.2614750 -0.1580463
## sample estimates:
## mean of x
## -1.709761
```

# Un-Paired Two Sample T-Test

- Need variance pooling, also the degrees of freedom has a funky formula
- Not going to talk about the details..

```
set.seed(2014)
sample3 = sample(pop, size = 111, replace = T) + rnorm(111, 0 ,0.11)
t.test(sample1, sample2, paired = F)
```

```
##
##  Welch Two Sample t-test
##
## data:  sample1 and sample2
## t = -2.2931, df = 95.811, p-value = 0.02403
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.1898512 -0.2296701
## sample estimates:
## mean of x mean of y
##  169.9741  171.6839
```

# Chi-Square Confidence Interval for Population Variance

- Make inference on the population variance
- Test statistic $\frac{(n-1)\hat{\sigma}^2}{{\sigma_0}^2} = \frac{\Sigma(X-\hat{\mu}_X)^2}{{\sigma_0}^2}$

```
myChisq.CI <- function(v, level = 0.95){
    n = length(v)
    left <- round((n-1)*var(v)/qchisq(1-(1-level)/2,n-1),4)
    right <- round((n-1)*var(v)/qchisq((1-level)/2,n-1),4)
    print(paste(left,right))
}
myChisq.CI(sample1)
var(pop)
```

```
## [1] "11.1645 24.8455"
## [1] 15.95734
```

- You can perform Chi-Square test on it too.

```
myChisq.Test <- function(v, var0, tail = "Two-tail"){
    n = length(v)
    chisq <- (n-1)*var(v)/var0
    if (tail == "Right-tail"){p <- 1-pchisq(chisq,n-1)}
    else if (tail == "Left-tail"){p <- pchisq(chisq,n-1)}
    else {
        if (var > var0) p <- 2*(1-pchisq(chisq, n-1))
        else p <- 2*(pchisq(chisq, n-1))}
    return(p)}
myChisq.Test(sample1,10, "Right-tail")
```

```
## [1] 0.004821433
```

```
var(sample1)
```

```
## [1] 15.99994
```

- Hypothesis for the ratio of two population variances
- $H_0 : \frac{{\sigma_1}^2}{{\sigma_2}^2} = 1$

```
myF.Test <- function(v1,v2,tail = "Two-tail"){
    var1 <- var(v1); var2 <- var(v2); n1 <- length(v1); n2 <- length(v2)
    f <- var1/var2
    if (tail == "Right-tail") p <- 1-pf(f, n1-1,n2-1)
    else if (tail == "Left-tail") p <- pf(f, n1-1,n2-1)
    else {
        if (var1 > var2) p <- 2*(1-pf(f, n1-1, n2-1))
        else p <- 2*(pf(f, n-1, n2-1))
    return(p)}
}
var(sample1);var(sample2);print(myF.Test(sample1, sample2, "Right-tail"))
```

```
## [1] 15.99994
## [1] 11.79786
## [1] 0.1448279
```

# Break

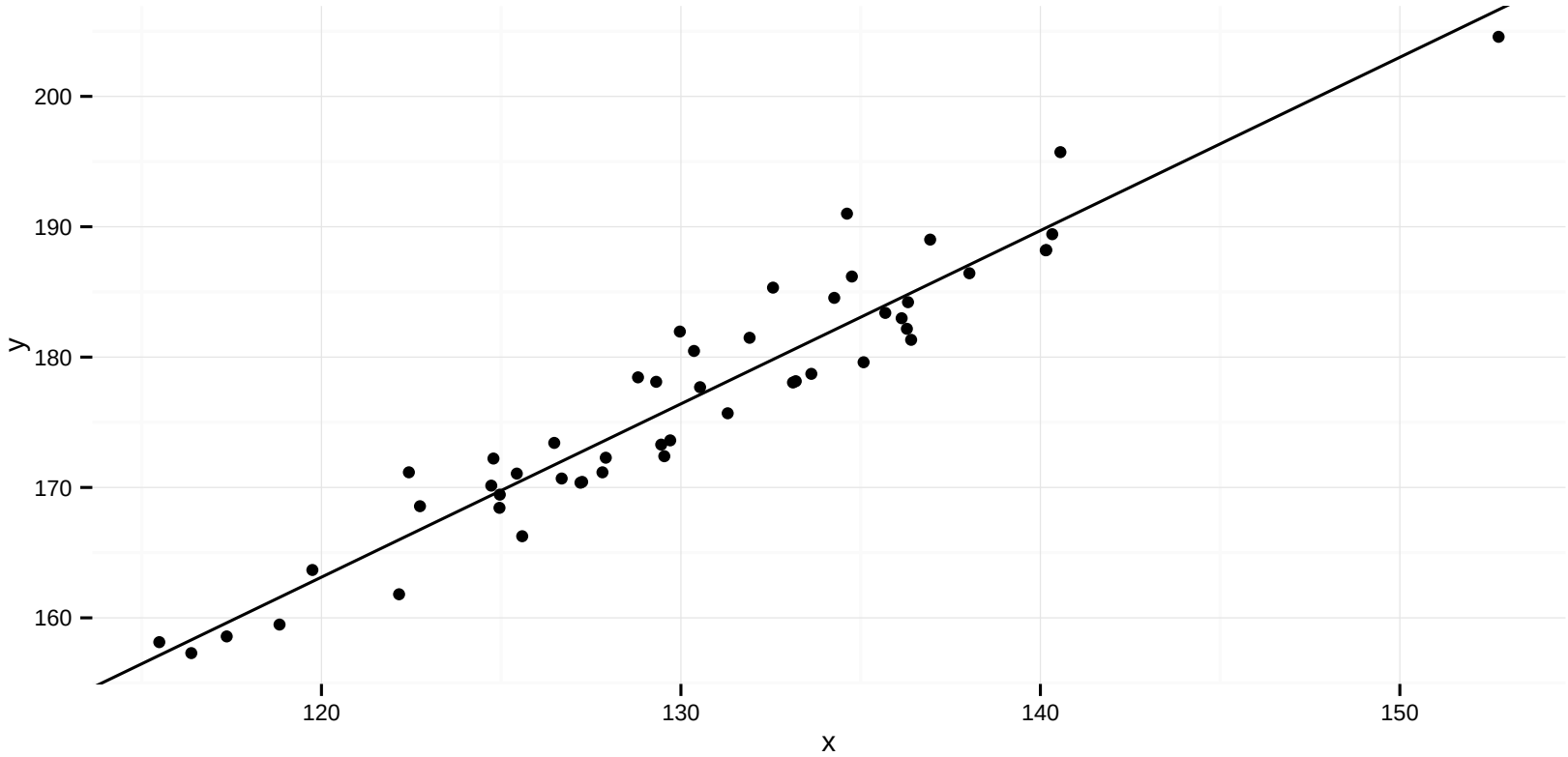
- I Guess GR521 didn't go this far now... Right?
- We switch to ST625 for now...

- Identify linear relationship between dependent variable and independent variables
- Linear function is of form :
  - $Y = \alpha + \beta X$
- Example data

```
set.seed(312)
weights <- rnorm(50, 130, 7)
heights <- weights * 1.3 + rpois(50,7)
df <- cbind.data.frame(y = heights, x = weights)
```

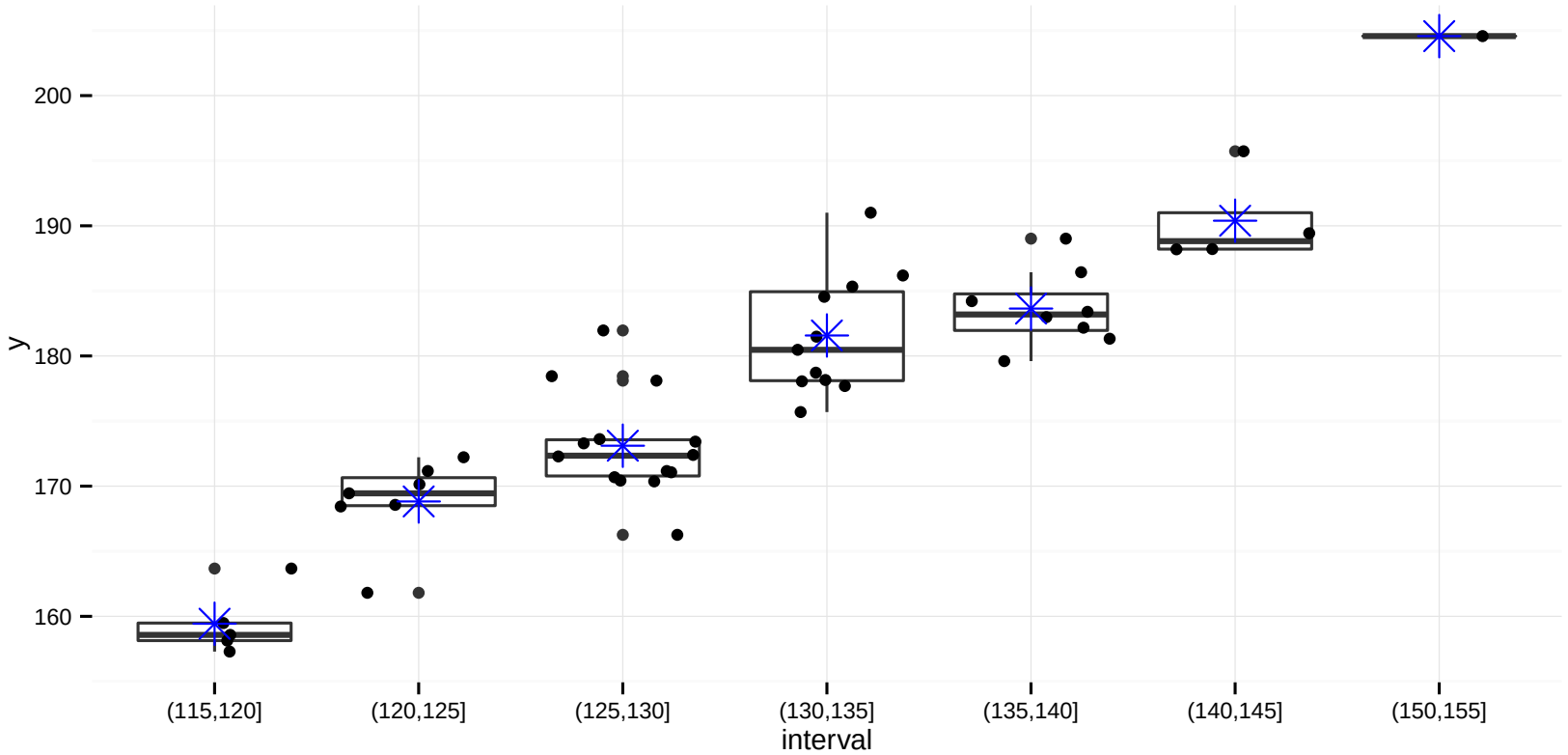
- One dependent variable in the linear function

```
qplot(x = x, y = y, data = df) +
    geom_abline(intercept = 3.598477, slope = 1.329348)
```

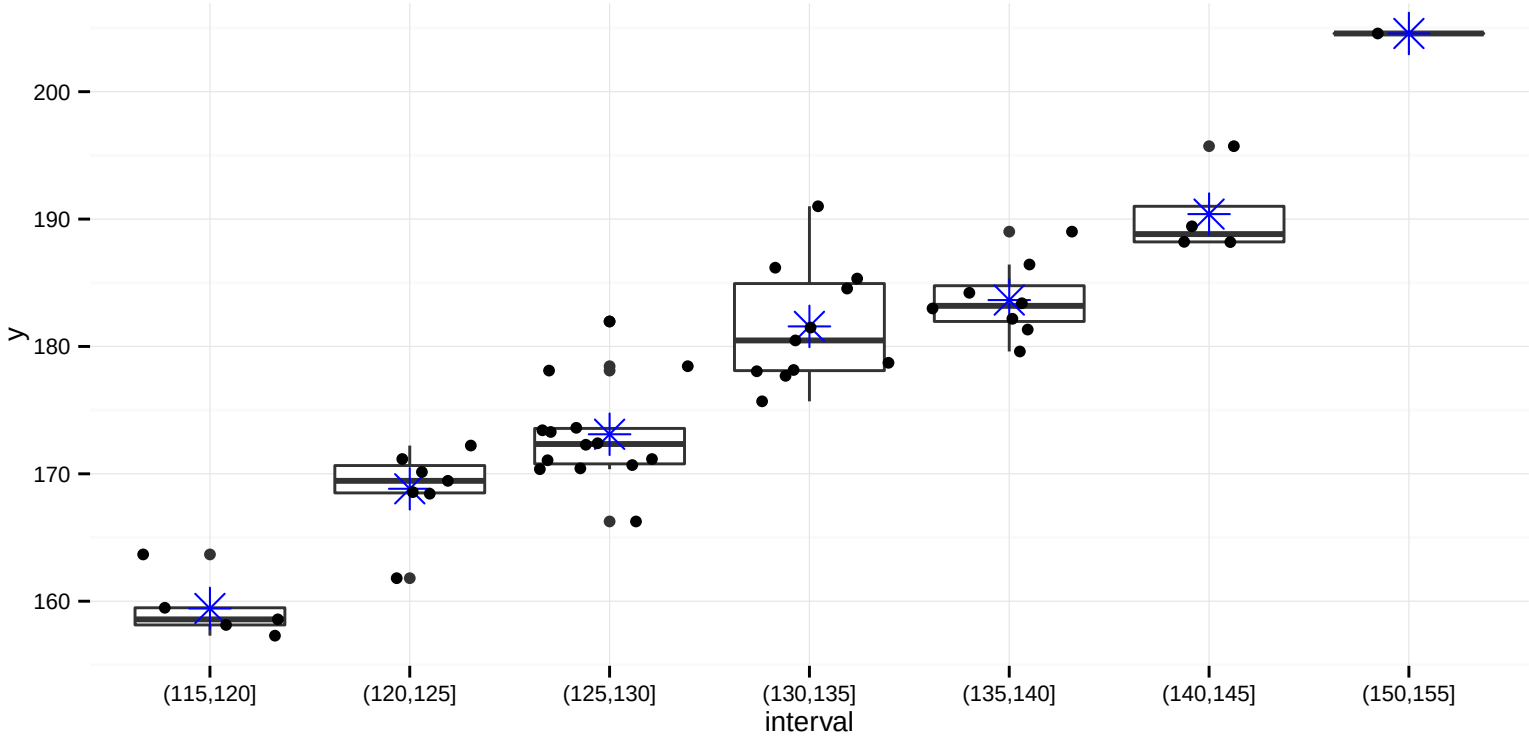


- Conditional Means: $\mu_{Y|X}$

```
df$interval <- cut(df$x, breaks = seq(115,155,5))
qplot(x = interval, y = y, data = df, geom = "boxplot") + geom_jitter() +
    stat_summary(fun.y=mean,  geom="point", shape=8, size=5,color = "blue")
```



1. Linearity: regression line connecting all conditional means
2. Nomarlity: all conditional distribution are normally distributed
3. Equal Variance (Homeoscedasticity): variances for all conditional distribution are the same
4. Independence of the error terms: residuals are independently distributed



# OLS Regression Line

- Find the coefficients for $Y = \alpha + \beta X$ such that the RMSE, MSE, SSE can be minimized
- SSE $= \Sigma(Y - \hat{Y})^2$

```
model <- lm(y~x, data= df)
coefficients <- summary(model)$coefficients[,"Estimate"]
coefficients
RMSE <- sqrt(sum(model$residuals^2)/model$df.residual)
RMSE
```

```
## (Intercept)           x
##    3.598477    1.329348
## [1] 2.986083
```

```
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2841 -2.3455 -0.8654  2.2957  8.4508
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.59848    7.74260   0.465    0.644
## x            1.32935    0.05937  22.390   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.986 on 48 degrees of freedom
## Multiple R-squared:  0.9126, Adjusted R-squared:  0.9108
## F-statistic: 501.3 on 1 and 48 DF,  p-value: < 2.2e-16
```

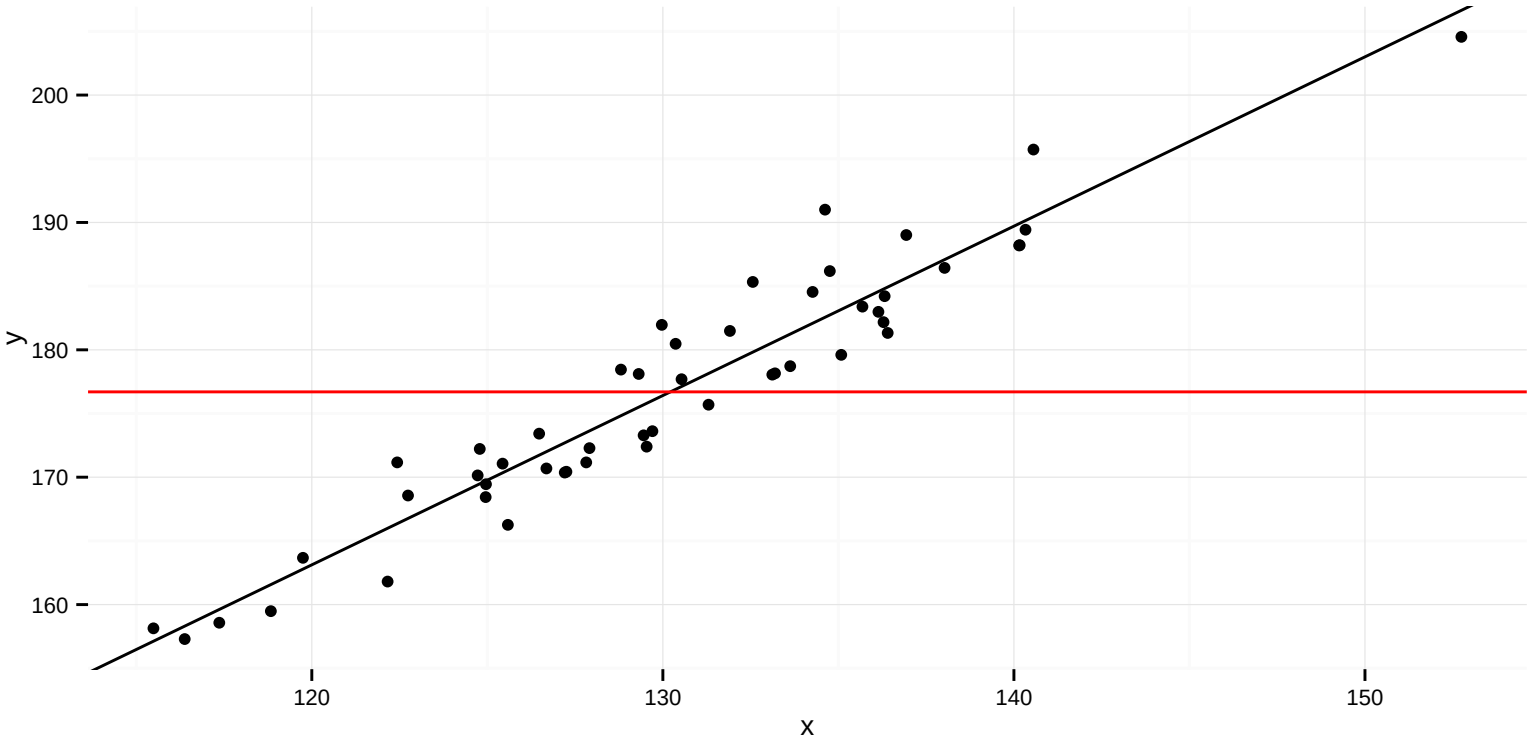
# Make Predictions

- This is calculate the estimated values for conditional means
- $\hat{Y} = a \times 1 + bX$
- Where $a$ and $b$ are estimated values for $\alpha$ and $\beta$ from the OLS fitted linear regression function

```
c(1, 144) %*% coefficients
model$fitted.values[1:5]
```

```
##          [,1]
## [1,] 195.0245
##        1        2        3        4        5
## 183.9710 190.4468 184.9275 169.4796 175.4987
```

- Use a horizontal line at height of $\bar{Y}$ as predictions
- RMSE for OLS line 2.9860834
- RMSE for horizontal overall mean line 9.9981528 , this is simply the standard deviation of y



# Sampling Distribution of the Slope Coefficient

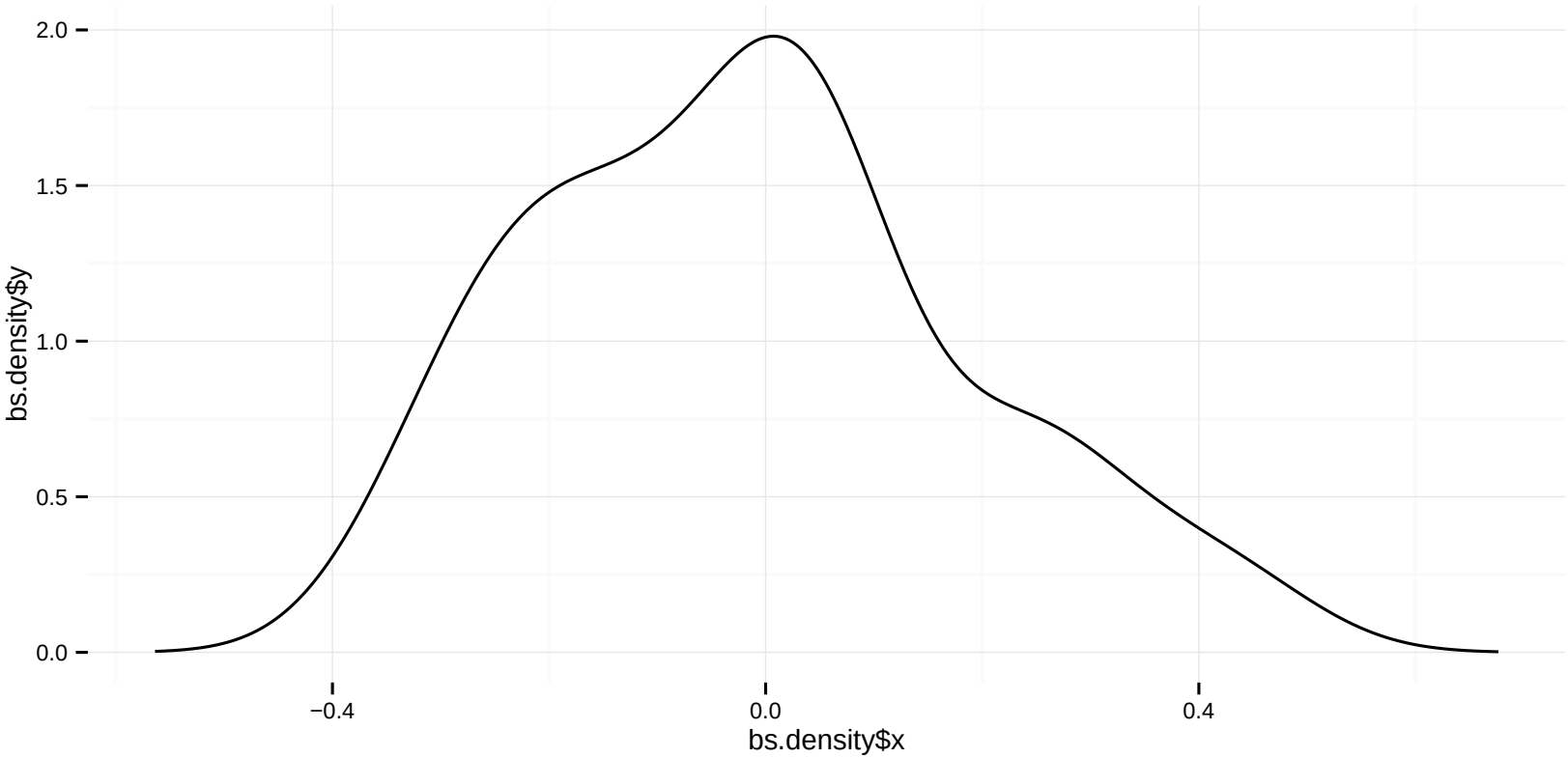
- Similar to the sampling distribution of mean
- If we can drawn many independent samples from the population
- For each sample we get the coefficients via OLS
- Then we get a sampling distribution of both $a$ and $b$

```
bs <- vector() -> as
set.seed(36)
for (i in 1:30){
    w <- rnorm(50, 130, 7); h <- weights * 1.3 + rpois(50,7);m <- lm(h~w);
    as <- c(as,m$coefficients[1])
    bs <- c(bs,m$coefficients[2])}
bs[1:5]
```

```
##           w           w           w           w           w
## -0.25762467 -0.01142427 -0.25534100  0.04347458  0.00399665
```

# Sampling Distribution of the Slope Coefficient

```r
bs.density <- density(bs)
qplot(bs.density$x, bs.density$y, geom = "line")
```



# Standard Error for Sampling Distribution of the Slope Coefficient

```r
b.SE <- RMSE/sqrt(sum((df$x - mean(df$x))^2))
summary(model)
b.SE
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2841 -2.3455 -0.8654  2.2957  8.4508
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.59848    7.74260   0.465    0.644
## x            1.32935    0.05937  22.390   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.986 on 48 degrees of freedom
## Multiple R-squared:  0.9126, Adjusted R-squared:  0.9108
## F-statistic: 501.3 on 1 and 48 DF,  p-value: < 2.2e-16
##
## [1] 0.05937144
```

# T-Test for the Slope Coefficient

- $\frac{(b-\beta)}{\hat{\sigma}_b}$ is t distributed

```
t <- coefficients[2]/b.SE
pt(t, df = model$df.residual, lower.tail = F)
summary(model)$coefficients
```

```
##              x
## 2.348994e-27
##              Estimate Std. Error    t value     Pr(>|t|)
## (Intercept) 3.598477 7.74259940  0.4647634 6.442024e-01
## x           1.329348 0.05937144 22.3903559 4.697989e-27
```

# Sampling Distribution of Estimated Conditional Mean

- Use the different regression functions we got using sampling method, we can make many predictions for a given x value
- These predictions are the estimated conditional mean that different regression line pass through
- And... these conditional means form a sampling distribution

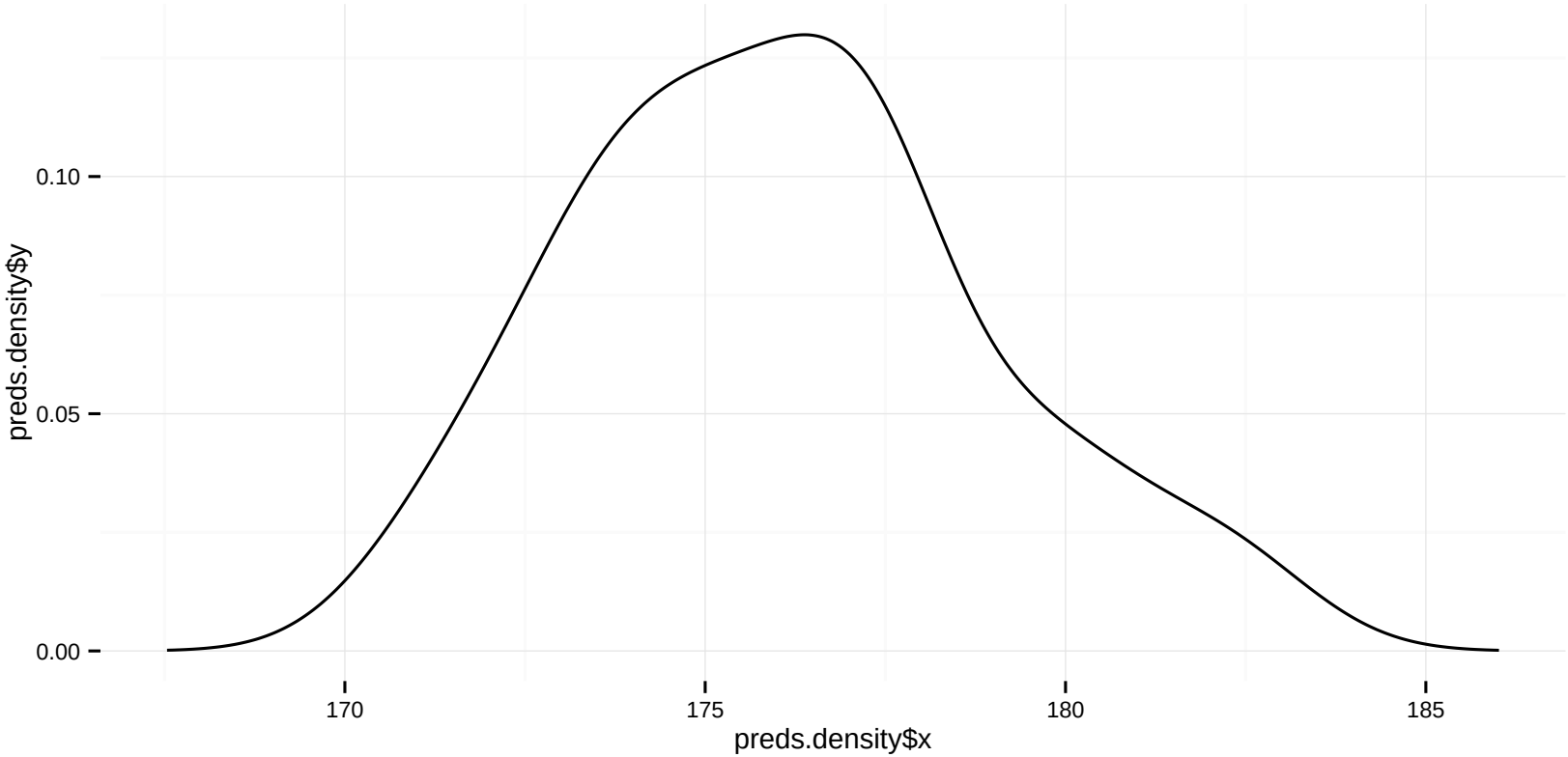
```
preds <- cbind(as, bs) %*% c(1,144)
row.names(preds) <- NULL
preds[1:5]
```

```
## [1] 173.0738 176.6668 173.6359 176.6817 175.7985
```

# Sampling Distribution of Estimated Conditional Mean

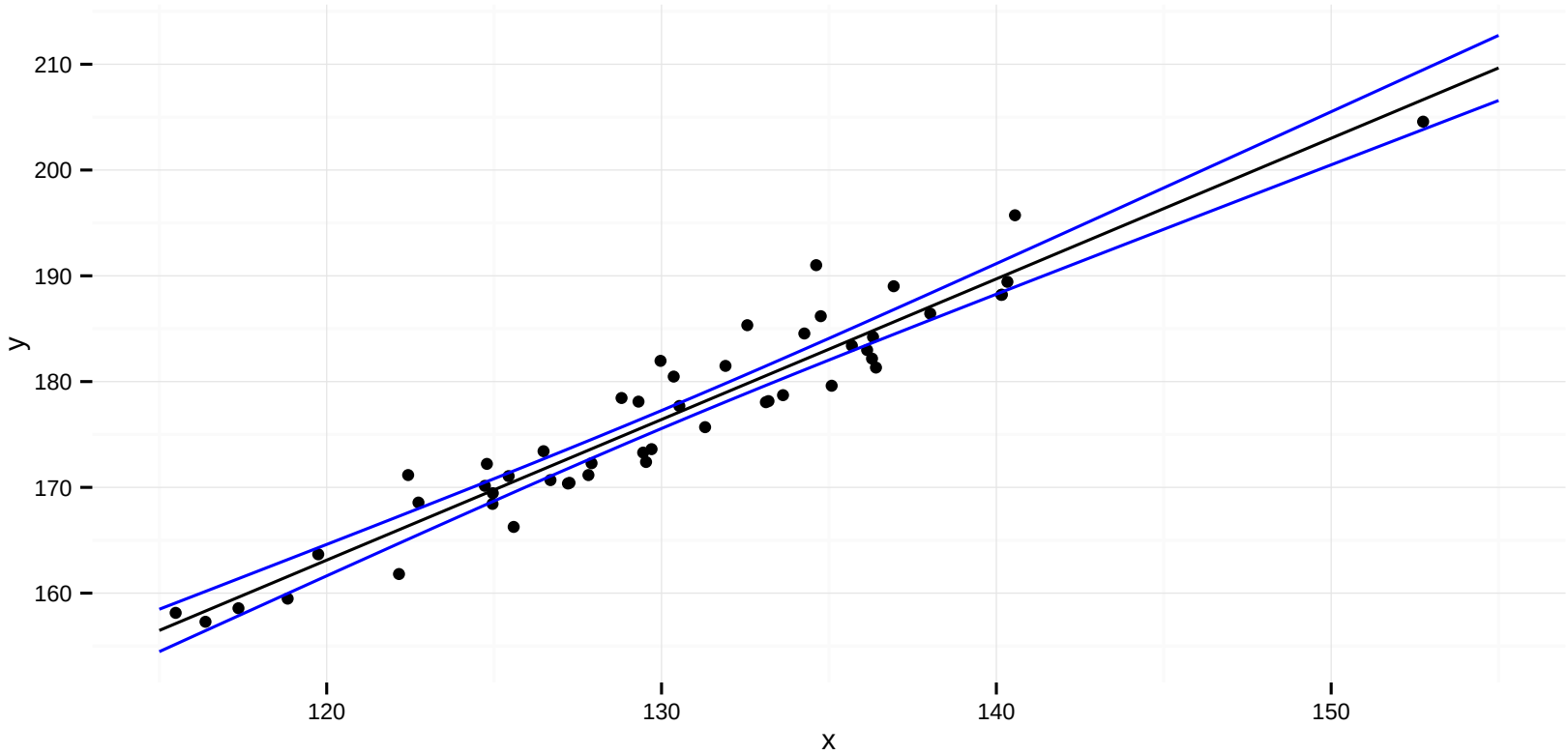
- The standard error term for this sampling distribution is funky
- $\sigma_{\hat{\mu}_{Y|X}} = \sqrt{\frac{\sigma^2_{Y|X}}{n} + (X - \bar{X})^2 \frac{\sigma^2_{Y|X}}{\Sigma(X-\bar{X})^2}}$

```
preds.density <- density(preds)
qplot(preds.density$x, preds.density$y, geom = "line")
```



# Confidence Interval of the Predictions

```
weight.linspace <- seq(115,155,1)
predict.CI <- predict(model, data.frame("x" = weight.linspace), interval = "con
```

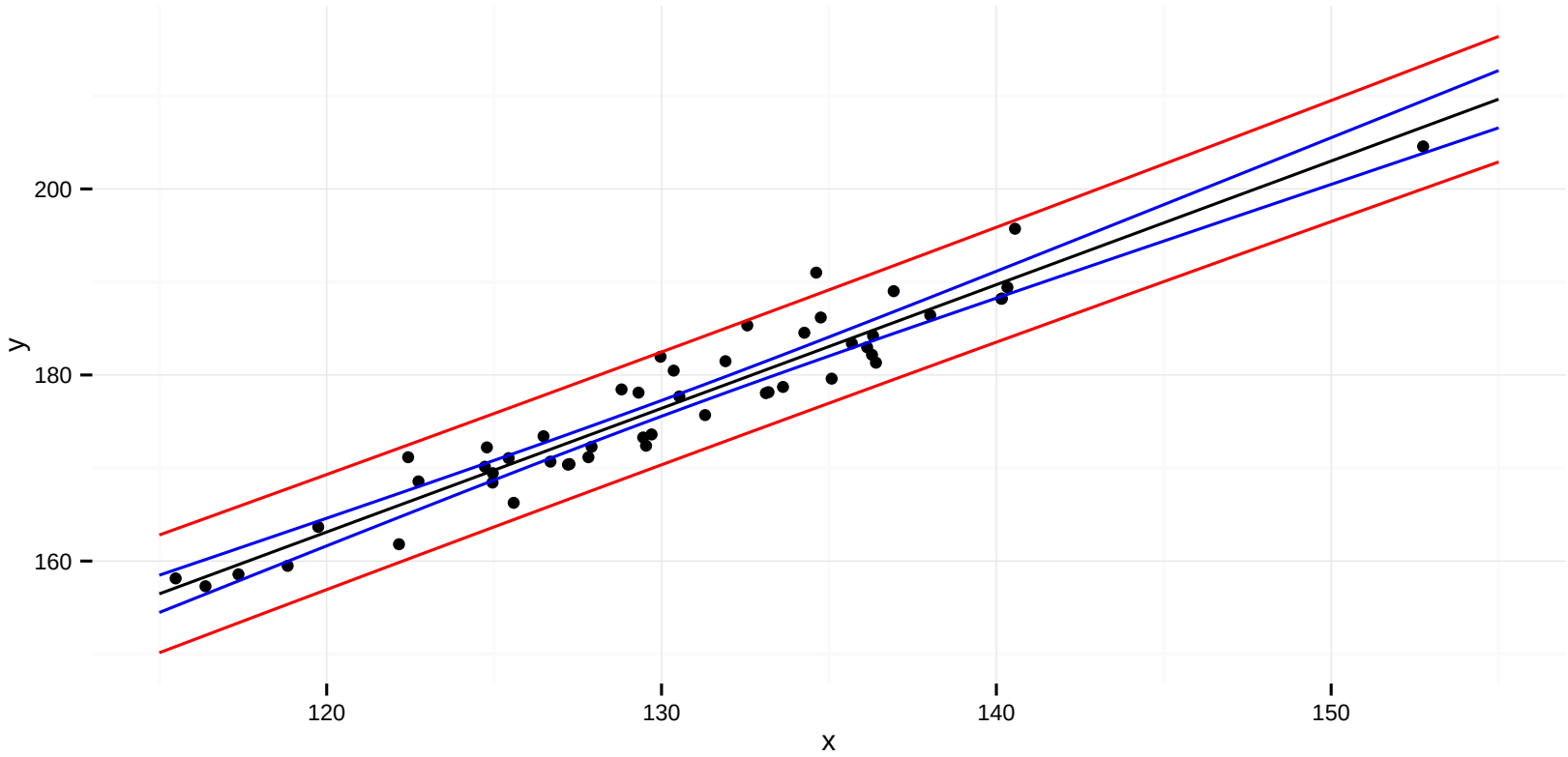


# Prediction Interval

- Standard Error is even more funky:

$$\sigma_{\hat{\mu}_{Y|X}} = \sqrt{\sigma^2_{Y|X} + \frac{\sigma^2_{Y|X}}{n} + (X - \bar{X})^2 \frac{\sigma^2_{Y|X}}{\Sigma(X - \bar{X})^2}}$$

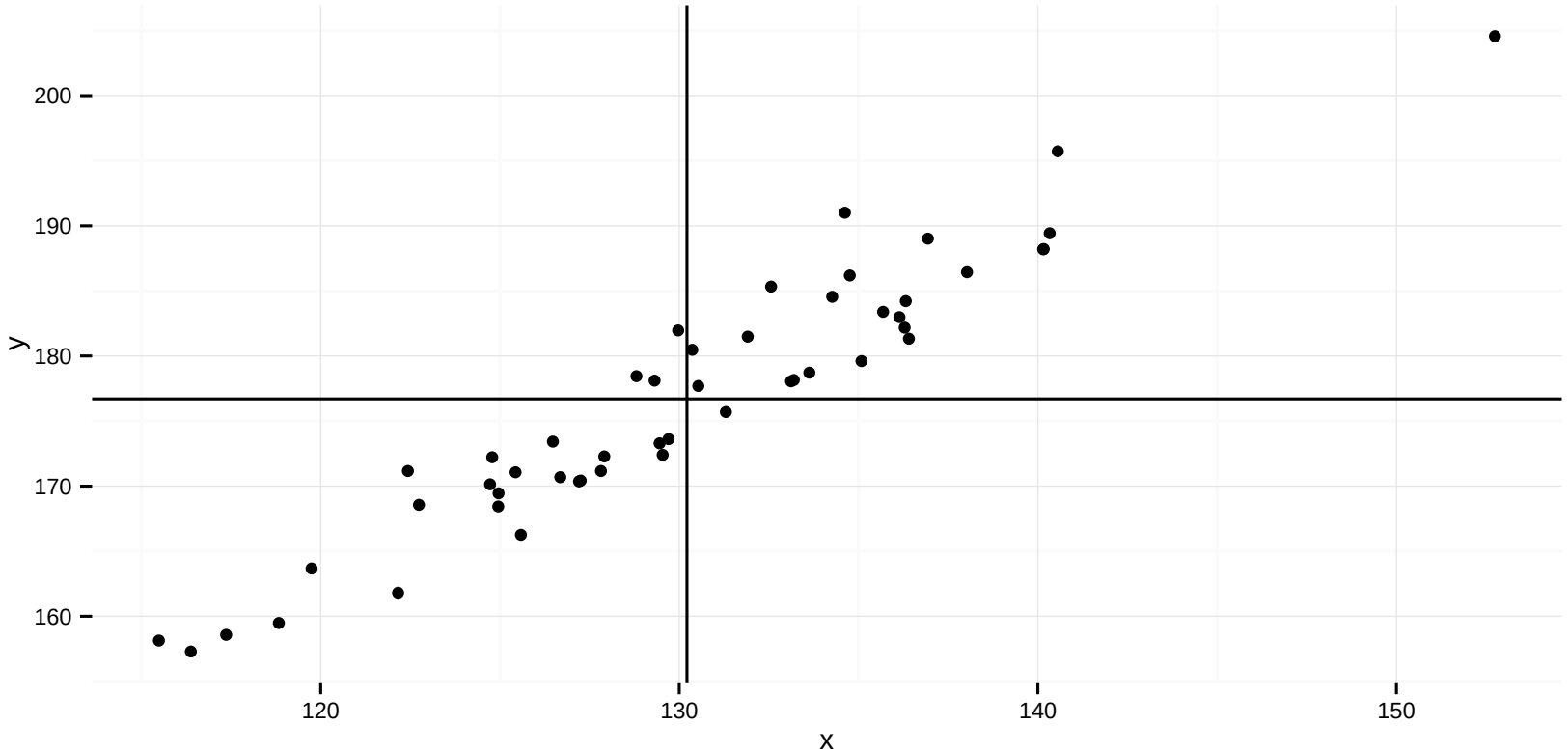
```
predict.PI <- predict(model, data.frame("x" = weight.linspace), interval = "pre
```



# Covariance

- $\frac{\Sigma(X-\bar{X})(Y-\bar{Y})}{n}$

```
ggplot(aes(x = x, y = y), data = df) + geom_point() +
    geom_vline(xintercept = mean(df$x)) +
    geom_hline(yintercept = mean(df$y))
```



# Coefficient of Correlation

- Coefficient of Correlation is standardized covariance
- $\frac{\Sigma(X-\bar{X})(Y-\bar{Y})}{n\sigma_X\sigma_Y}$
- It has value within range [-1, 1]
- The absolute value of it suggest the strength of linear relationship

```
cov(df$x, df$y)/sd(df$x)/sd(df$y)
cor(df$x, df$y)
```

```
## [1] 0.9553117
## [1] 0.9553117
```

# Coefficient of Determination

- Total bariability in Y = variability associated with X + variability not associated with X

$$\hat{\sigma}_Y^2 = \hat{\sigma}_{\mu_{Y|X}}^2 + \hat{\sigma}_{Y|X}^2$$

$$\frac{\Sigma(Y - \hat{\mu}_Y)^2}{n-1} = \frac{\Sigma(\hat{\mu}_{Y|X} - \hat{\mu}_Y)^2}{n-1} + \frac{\Sigma(Y - \hat{\mu}_{Y|X})^2}{n-1}$$

```
sum((df$y - mean(df$y))^2)/(length(df$y)-1)
sum((model$fitted.values-mean(df$y))^2)/(length(df$y)-1)
sum((df$y - model$fitted.values)^2)/(length(df$y)-1)
91.22834+8.73472
1-8.73472/99.96306
cor(df$y, df$x)^2
```

```
## [1] 99.96306
## [1] 91.22834
## [1] 8.73472
## [1] 99.96306
## [1] 0.9126205
## [1] 0.9126205
```

# Adjusted R Squared

- $R^2_{adj} = 1 - \frac{n-1}{n-k-1}(1 - R^2)$

```
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2841 -2.3455 -0.8654  2.2957  8.4508
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.59848    7.74260   0.465    0.644
## x            1.32935    0.05937  22.390   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.986 on 48 degrees of freedom
## Multiple R-squared:  0.9126, Adjusted R-squared:  0.9108
## F-statistic: 501.3 on 1 and 48 DF,  p-value: < 2.2e-16
```

- Partition of sum of squares again...
- Sum of square deviations = sum of square of regression + sum of square of errors
- $$\Sigma(Y - \mu_Y)^2 = \Sigma(\hat{\mu}_{Y|X} - \mu_Y)^2 + \Sigma(Y - \hat{\mu}_{Y|X})^2$$

```
SSR <- sum((model$fitted.values-mean(df$y))^2)
SSE <- sum((df$y-model$fitted.values)^2)
SS <- sum((df$y-mean(df$y))^2)
SSR + SSE
SS
```

```
## [1] 4898.19
## [1] 4898.19
```

- $F = \frac{\text{Mean Square Regression}}{\text{Mean Square Error}}$
-

$$F = \frac{\frac{\Sigma(\hat{\mu}_{Y|X} - \mu_Y)^2}{k-1}}{\frac{\Sigma(Y - \hat{\mu}_{Y|X})^2}{n-1}}$$

```
MSR <- sum((model$fitted.values-mean(df$y))^2)/(2-1)
MSE <- sum((df$y-model$fitted.values)^2)/model$df.residual
f <- MSR/MSE
c(MSR,MSE,f,pf(f,1,48,lower.tail = F))
anova(model)
```

```
## [1] 4.470189e+03 8.916694e+00 5.013280e+02 4.697989e-27
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1 4470.2  4470.2  501.33 < 2.2e-16 ***
## Residuals 48  428.0     8.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- What is been tested?
- Test whether $\mu_{Y|X} = \mu_Y$
- Test whether $\beta_0 = \beta_1 = ... = \beta_m = 0$
- Test whether $R^2 = 0$
- Test whether $\rho = 0$
- Same thing...

# Model Comparison Using Anova

```
model2 <- lm(y~x+I(x^2), data =df)
anova(model2, model)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x + I(x^2)
## Model 2: y ~ x
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     47 423.44
## 2     48 428.00 -1   -4.5657 0.5068 0.4801
```