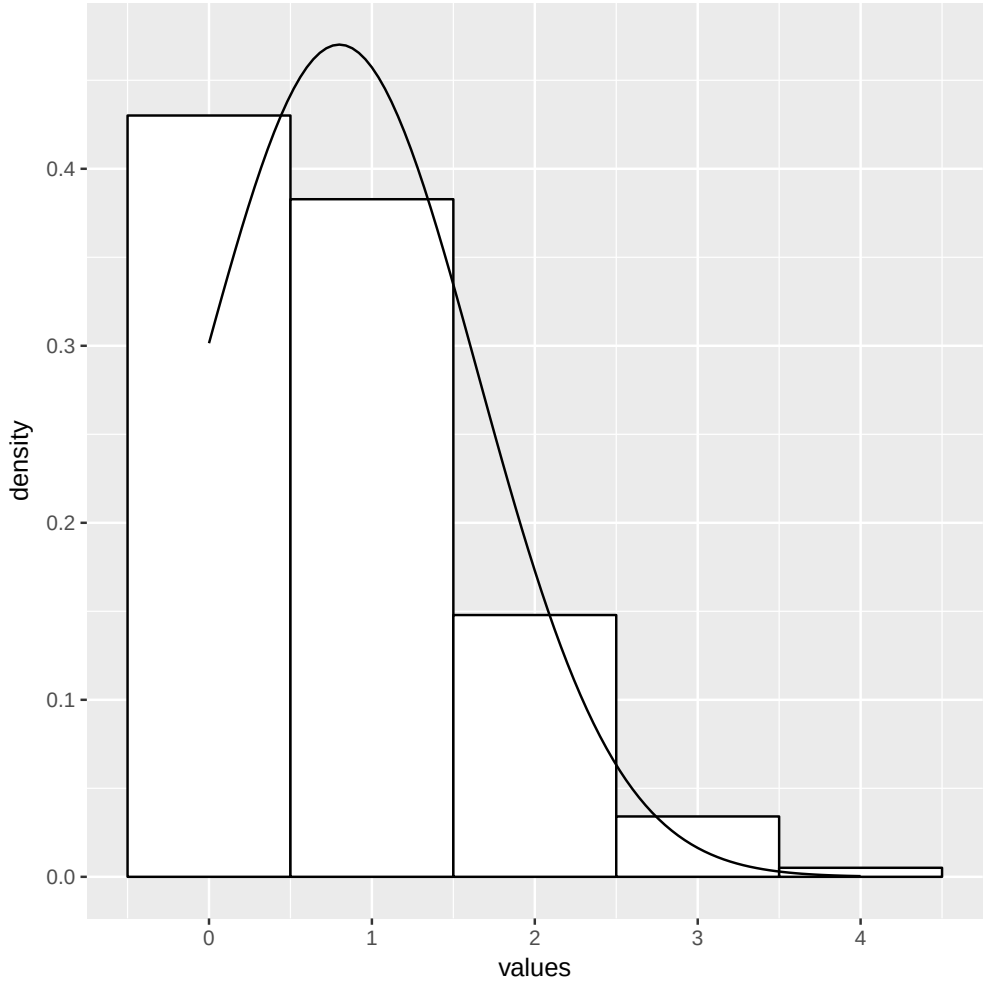
Statistics 3080
Homework 11
David Smith

Problem 1a

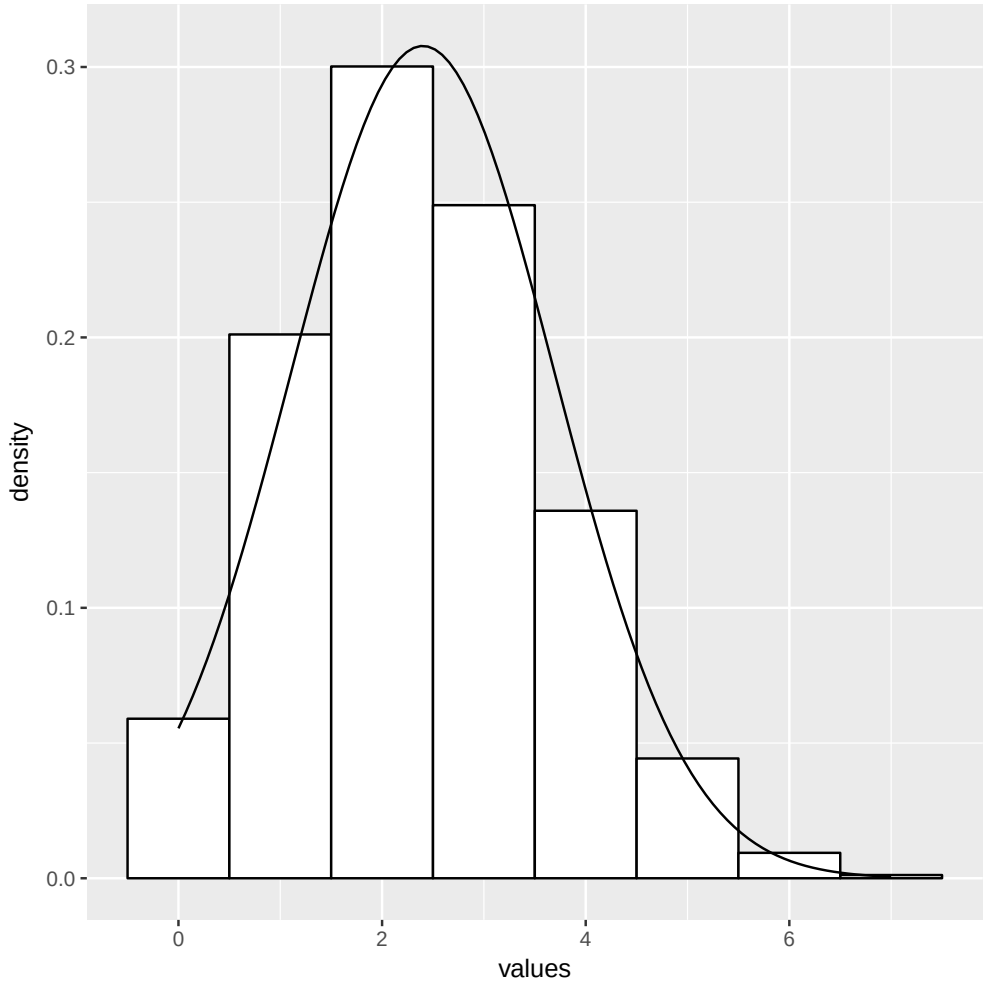
```
> library(ggplot2)
> set.seed(7311986)
> bin_a <- data.frame(values=rbinom(10000, 8, 0.1))
> mu_a <- 8*0.1
> sd_a <- sqrt(8*0.1*(1-0.1))
> ggplot(bin_a) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_a, sd=sd_a))
```



Problem 1b

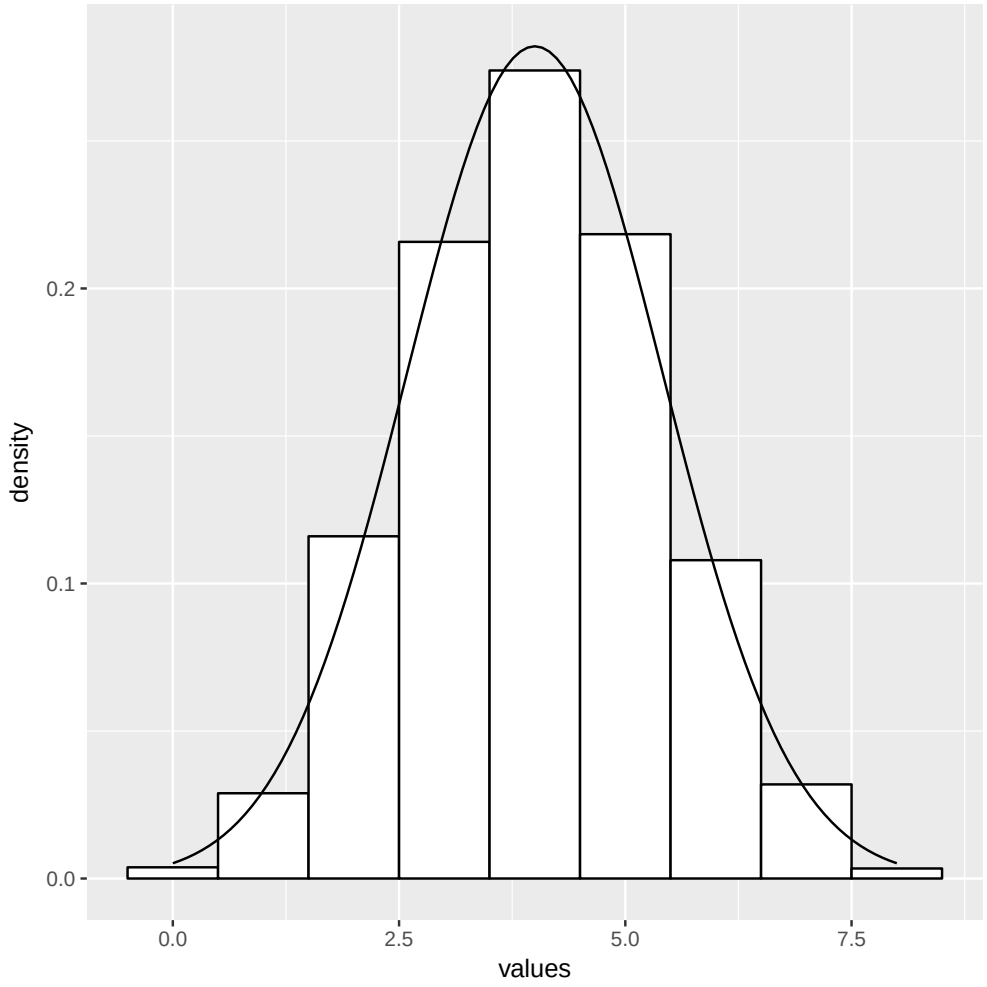
```
> bin_b <- data.frame(values=rbinom(10000, 8, 0.3))
> mu_b <- 8*0.3
```

```
> sd_b <- sqrt(8*0.3*(1-0.3))
> ggplot(bin_b) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_b, sd=sd_b))
```



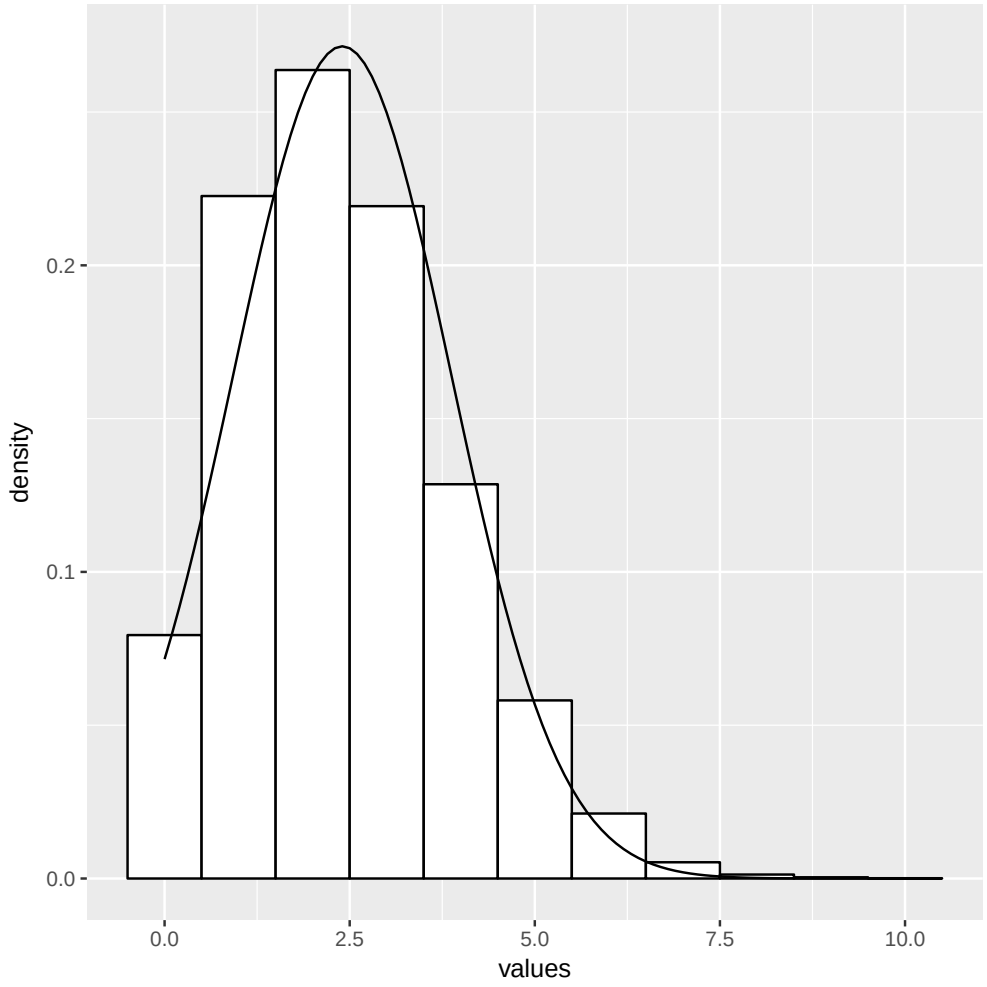
Problem 1c

```
> bin_c <- data.frame(values=rbinom(10000, 8, 0.5))
> mu_c <- 8*0.5
> sd_c <- sqrt(8*0.5*(1-0.5))
> ggplot(bin_c) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_c, sd=sd_c))
```



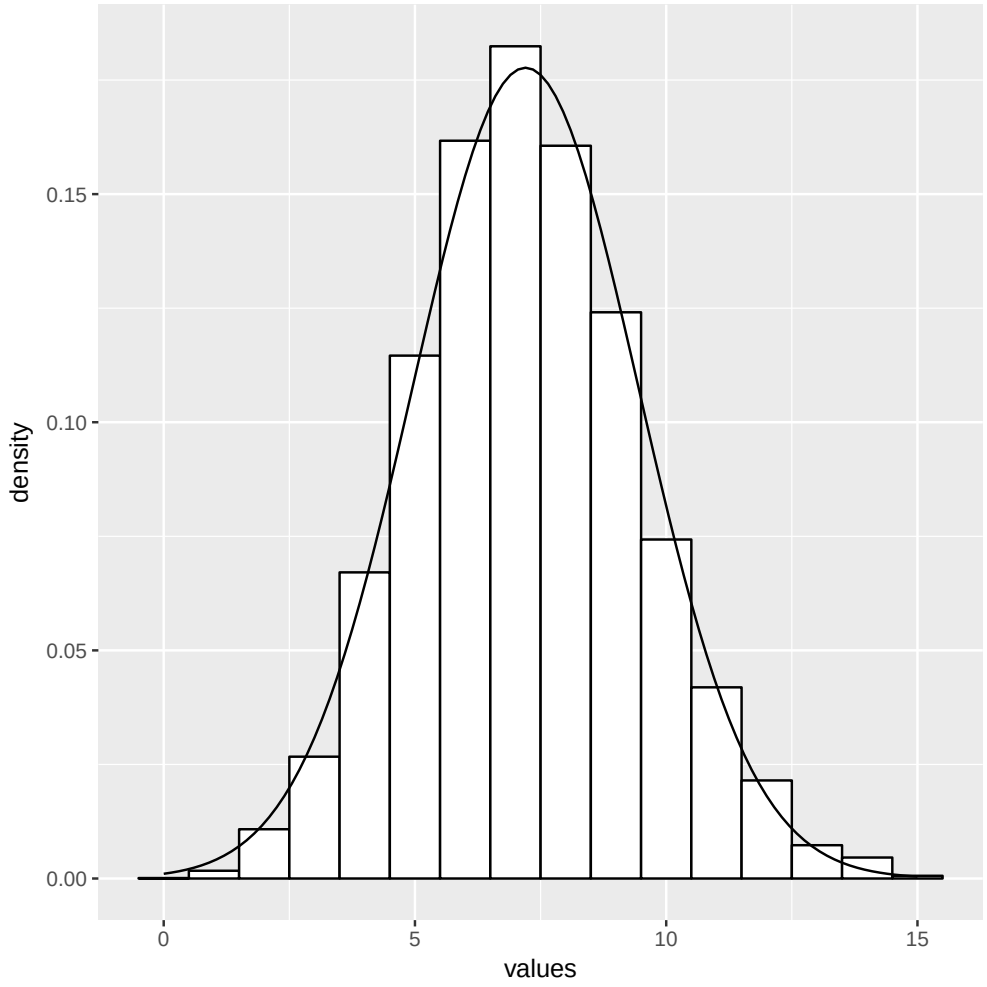
Problem 1d

```
> bin_d <- data.frame(values=rbinom(10000, 24, 0.1))
> mu_d <- 24*0.1
> sd_d <- sqrt(24*0.1*(1-0.1))
> ggplot(bin_d) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_d, sd=sd_d))
```



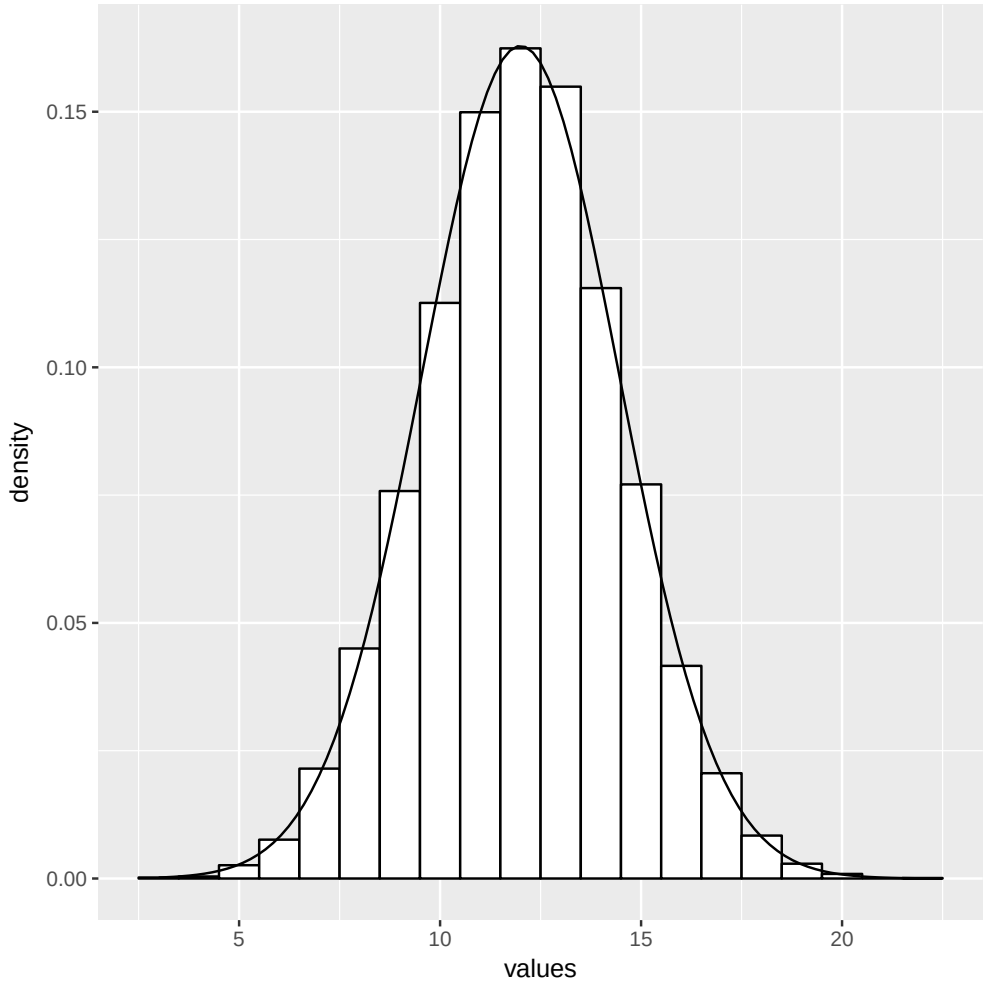
Problem 1e

```
> bin_e <- data.frame(values=rbinom(10000, 24, 0.3))
> mu_e <- 24*0.3
> sd_e <- sqrt(24*0.3*(1-0.3))
> ggplot(bin_e) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_e, sd=sd_e))
```



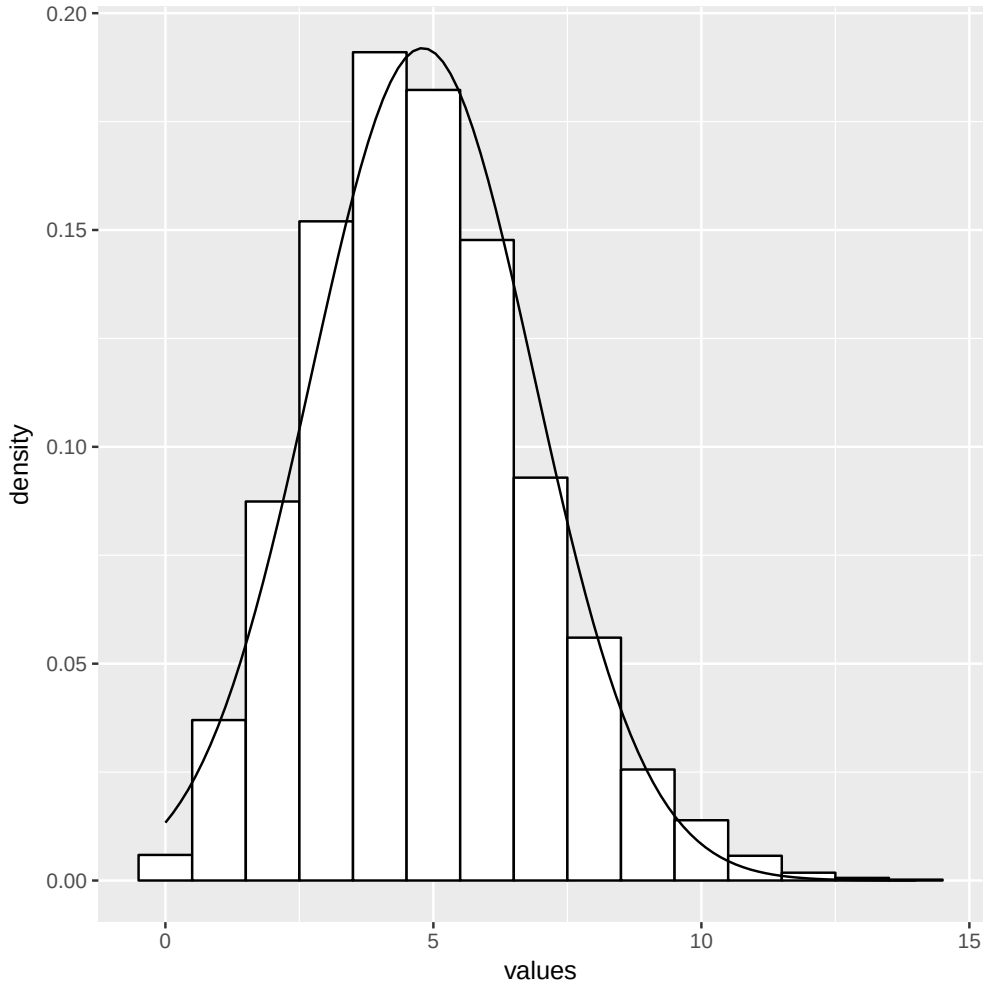
Problem 1f

```
> bin_f <- data.frame(values=rbinom(10000, 24, 0.5))
> mu_f <- 24*0.5
> sd_f <- sqrt(24*0.5*(1-0.5))
> ggplot(bin_f) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_f, sd=sd_f))
```



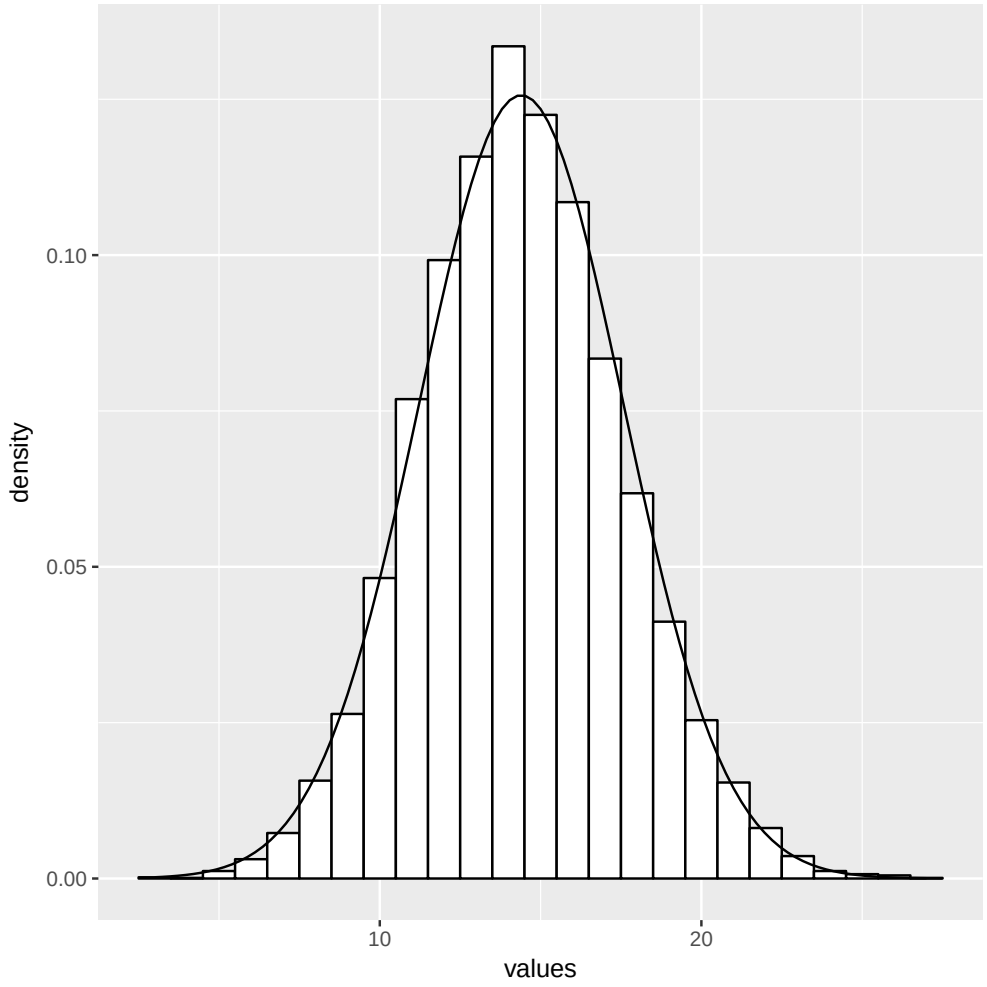
Problem 1g

```
> bin_g <- data.frame(values=rbinom(10000, 48, 0.1))
> mu_g <- 48*0.1
> sd_g <- sqrt(48*0.1*(1-0.1))
> ggplot(bin_g) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_g, sd=sd_g))
```



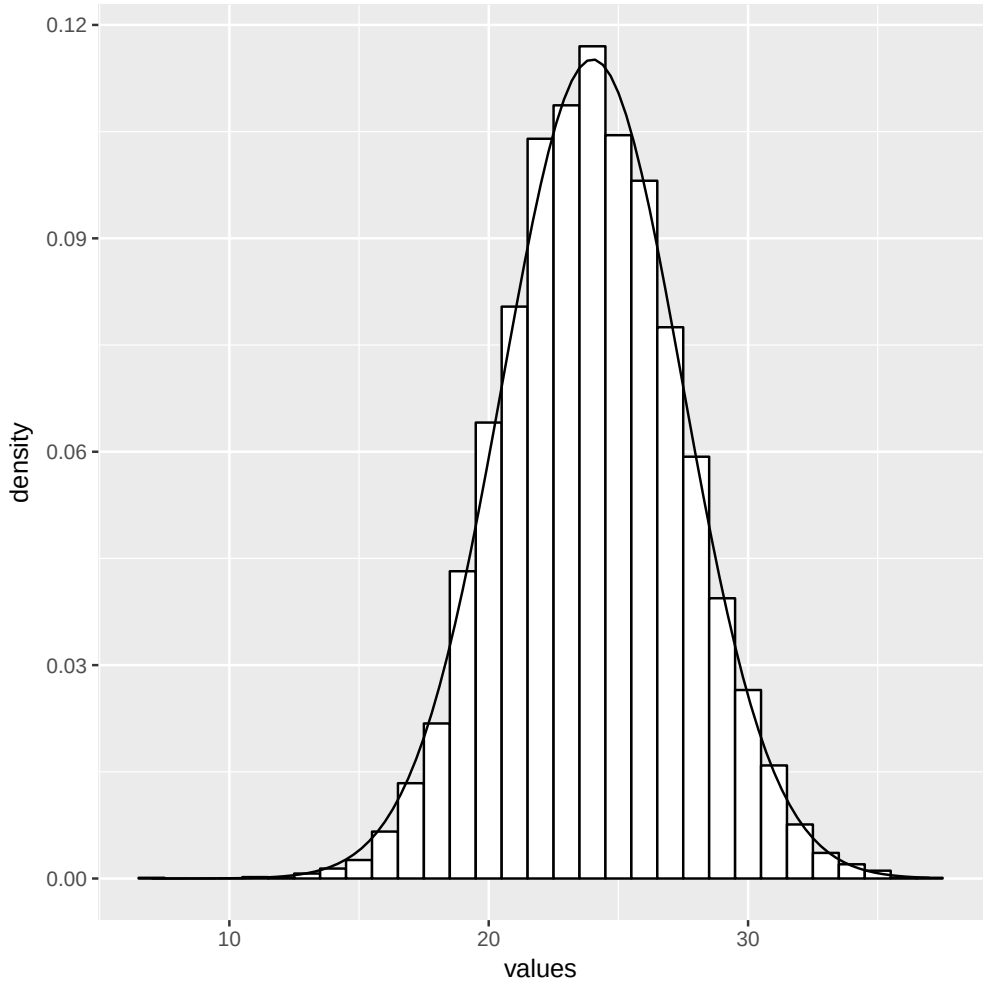
Problem 1h

```
> bin_h <- data.frame(values=rbinom(10000, 48, 0.3))
> mu_h <- 48*0.3
> sd_h <- sqrt(48*0.3*(1-0.3))
> ggplot(bin_h) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_h, sd=sd_h))
```



Problem 1i

```
> bin_i <- data.frame(values=rbinom(10000, 48, 0.5))
> mu_i <- 48*0.5
> sd_i <- sqrt(48*0.5*(1-0.5))
> ggplot(bin_i) + geom_histogram(aes(x=values, y=..density..),
+         colour="black", fill="white", binwidth=1) +
+         stat_function(fun=dnorm, args=list(mean=mu_i, sd=sd_i))
```



Problem 2

```
> print("The normal approximation is generally more valid when n is")
[1] "The normal approximation is generally more valid when n is"
> print("large, and when p is not small. It seems best for distributions")
[1] "large, and when p is not small. It seems best for distributions"
> print("e, f, g, h, and i, and is not so good for distributions a, b, c,")
[1] "e, f, g, h, and i, and is not so good for distributions a, b, c,"
> print("and d. When p is small, the distributions are skewed right, and")
[1] "and d. When p is small, the distributions are skewed right, and"
> print("when n is small, they are very discrete, and the normal curve may")
```

```
[1] "when n is small, they are very discrete, and the normal curve may"

> print("over or underestimate.")

[1] "over or underestimate."
```

Problem 3

```
> prop_a <- lapply(bin_a$values, FUN=prop.test, n=8, p=0.1,
+                  alternative="two.sided", correct=FALSE)
> p_a <- do.call(rbind, lapply(prop_a, function(z) {z$p.value}))
> reject_a <- p_a < 0.05
> t1_a <- mean(reject_a)
> prop_b <- lapply(bin_b$values, FUN=prop.test, n=8, p=0.3,
+                  alternative="two.sided", correct=FALSE)
> p_b <- do.call(rbind, lapply(prop_b, function(z) {z$p.value}))
> reject_b <- p_b < 0.05
> t1_b <- mean(reject_b)
> prop_c <- lapply(bin_c$values, FUN=prop.test, n=8, p=0.5,
+                  alternative="two.sided", correct=FALSE)
> p_c <- do.call(rbind, lapply(prop_c, function(z) {z$p.value}))
> reject_c <- p_c < 0.05
> t1_c <- mean(reject_c)
> prop_d <- lapply(bin_d$values, FUN=prop.test, n=24, p=0.1,
+                  alternative="two.sided", correct=FALSE)
> p_d <- do.call(rbind, lapply(prop_d, function(z) {z$p.value}))
> reject_d <- p_d < 0.05
> t1_d <- mean(reject_d)
> prop_e <- lapply(bin_e$values, FUN=prop.test, n=24, p=0.3,
+                  alternative="two.sided", correct=FALSE)
> p_e <- do.call(rbind, lapply(prop_e, function(z) {z$p.value}))
> reject_e <- p_e < 0.05
> t1_e <- mean(reject_e)
> prop_f <- lapply(bin_f$values, FUN=prop.test, n=24, p=0.5,
+                  alternative="two.sided", correct=FALSE)
> p_f <- do.call(rbind, lapply(prop_f, function(z) {z$p.value}))
> reject_f <- p_f < 0.05
> t1_f <- mean(reject_f)
> prop_g <- lapply(bin_g$values, FUN=prop.test, n=48, p=0.1,
+                  alternative="two.sided", correct=FALSE)
> p_g <- do.call(rbind, lapply(prop_g, function(z) {z$p.value}))
> reject_g <- p_g < 0.05
> t1_g <- mean(reject_g)
> prop_h <- lapply(bin_h$values, FUN=prop.test, n=48, p=0.3,
+                  alternative="two.sided", correct=FALSE)
```

```
> p_h <- do.call(rbind, lapply(prop_h, function(z) {z$p.value}))
> reject_h <- p_h < 0.05
> t1_h <- mean(reject_h)
> prop_i <- lapply(bin_i$values, FUN=prop.test, n=48, p=0.5,
+                  alternative="two.sided", correct=FALSE)
> p_i <- do.call(rbind, lapply(prop_i, function(z) {z$p.value}))
> reject_i <- p_i < 0.05
> t1_i <- mean(reject_i)
> table_p <- matrix(c(t1_a, t1_b, t1_c, t1_d, t1_e, t1_f, t1_g,
+            t1_h, t1_i), ncol=1)
> rownames(table_p) <- c("n=8,p=0.1", "n=8,p=0.3", "n=8,p=0.5",
+                        "n=24,p=0.1", "n=24,p=0.3", "n=24,p=0.5",
+                        "n=48,p=0.1", "n=48,p=0.3", "n=48,p=0.5")
> colnames(table_p) <- "Type I error"
> table_p

           Type I error
n=8,p=0.1        0.0392
n=8,p=0.3        0.0549
n=8,p=0.5        0.0680
n=24,p=0.1       0.0283
n=24,p=0.3       0.0466
n=24,p=0.5       0.0652
n=48,p=0.1       0.0537
n=48,p=0.3       0.0572
n=48,p=0.5       0.0555
```

Problem 4

```
> print("We see that the type I errors are the closest to 0.05 when n is")

[1] "We see that the type I errors are the closest to 0.05 when n is"

> print("large, and are generally worse when p is small, which is")

[1] "large, and are generally worse when p is small, which is"

> print("consistent with the plots. Therefore, we conclude that the")

[1] "consistent with the plots. Therefore, we conclude that the"

> print("one-sample z test for proportions depends upon the data")

[1] "one-sample z test for proportions depends upon the data"

> print("following an approximate normal distribution. Our observation")
```

```
[1] "following an approximate normal distribution. Our observation"

> print("that the normal curve may under or overestimate when n is small")

[1] "that the normal curve may under or overestimate when n is small"

> print("was also accurate, as we see type I errors that are both too")

[1] "was also accurate, as we see type I errors that are both too"

> print("large and too small.")

[1] "large and too small."
```

Problem 5

```
> prop_a <- lapply(bin_a$values, FUN=prop.test, n=8, p=0.1,
+                  alternative="two.sided", correct=TRUE)
> p_a <- do.call(rbind, lapply(prop_a, function(z) {z$p.value}))
> reject_a <- p_a < 0.05
> t12_a <- mean(reject_a)
> prop_b <- lapply(bin_b$values, FUN=prop.test, n=8, p=0.3,
+                  alternative="two.sided", correct=TRUE)
> p_b <- do.call(rbind, lapply(prop_b, function(z) {z$p.value}))
> reject_b <- p_b < 0.05
> t12_b <- mean(reject_b)
> prop_c <- lapply(bin_c$values, FUN=prop.test, n=8, p=0.5,
+                  alternative="two.sided", correct=TRUE)
> p_c <- do.call(rbind, lapply(prop_c, function(z) {z$p.value}))
> reject_c <- p_c < 0.05
> t12_c <- mean(reject_c)
> prop_d <- lapply(bin_d$values, FUN=prop.test, n=24, p=0.1,
+                  alternative="two.sided", correct=TRUE)
> p_d <- do.call(rbind, lapply(prop_d, function(z) {z$p.value}))
> reject_d <- p_d < 0.05
> t12_d <- mean(reject_d)
> prop_e <- lapply(bin_e$values, FUN=prop.test, n=24, p=0.3,
+                  alternative="two.sided", correct=TRUE)
> p_e <- do.call(rbind, lapply(prop_e, function(z) {z$p.value}))
> reject_e <- p_e < 0.05
> t12_e <- mean(reject_e)
> prop_f <- lapply(bin_f$values, FUN=prop.test, n=24, p=0.5,
+                  alternative="two.sided", correct=TRUE)
> p_f <- do.call(rbind, lapply(prop_f, function(z) {z$p.value}))
> reject_f <- p_f < 0.05
```

```
> t12_f <- mean(reject_f)
> prop_g <- lapply(bin_g$values, FUN=prop.test, n=48, p=0.1,
+                  alternative="two.sided", correct=TRUE)
> p_g <- do.call(rbind, lapply(prop_g, function(z) {z$p.value}))
> reject_g <- p_g < 0.05
> t12_g <- mean(reject_g)
> prop_h <- lapply(bin_h$values, FUN=prop.test, n=48, p=0.3,
+                  alternative="two.sided", correct=TRUE)
> p_h <- do.call(rbind, lapply(prop_h, function(z) {z$p.value}))
> reject_h <- p_h < 0.05
> t12_h <- mean(reject_h)
> prop_i <- lapply(bin_i$values, FUN=prop.test, n=48, p=0.5,
+                  alternative="two.sided", correct=TRUE)
> p_i <- do.call(rbind, lapply(prop_i, function(z) {z$p.value}))
> reject_i <- p_i < 0.05
> t12_i <- mean(reject_i)
> table_p <- matrix(c(t1_a, t1_b, t1_c, t1_d, t1_e, t1_f, t1_g,
+                     t1_h, t1_i, t12_a, t12_b, t12_c, t12_d, t12_e,
+                     t12_f, t12_g, t12_h, t12_i), ncol=2)
> rownames(table_p) <- c("n=8,p=0.1", "n=8,p=0.3", "n=8,p=0.5",
+                        "n=24,p=0.1", "n=24,p=0.3", "n=24,p=0.5",
+                        "n=48,p=0.1", "n=48,p=0.3", "n=48,p=0.5")
> colnames(table_p) <- c("Type I error, no cont. corr.",
+                        "Type I error, cont. corr.")
> table_p

          Type I error, no cont. corr. Type I error, cont. corr.
n=8,p=0.1                       0.0392                    0.0392
n=8,p=0.3                       0.0549                    0.0106
n=8,p=0.5                       0.0680                    0.0072
n=24,p=0.1                      0.0283                    0.0283
n=24,p=0.3                      0.0466                    0.0251
n=24,p=0.5                      0.0652                    0.0231
n=48,p=0.1                      0.0537                    0.0281
n=48,p=0.3                      0.0572                    0.0261
n=48,p=0.5                      0.0555                    0.0262

> print("We see that the type I errors are now far too small, and")

[1] "We see that the type I errors are now far too small, and"

> print("so we are not rejecting the null hypothesis as often as we")

[1] "so we are not rejecting the null hypothesis as often as we"

> print("should. Doing some research, I found that the Yates continuity")
```

```
[1] "should. Doing some research, I found that the Yates continuity"
> print("correction is usually only used with small sample sizes, but")
[1] "correction is usually only used with small sample sizes, but"
> print("even then, it may overcorrect, leading to an overly conservative")
[1] "even then, it may overcorrect, leading to an overly conservative"
> print("conclusion. Since we have an extremely large sample size of")
[1] "conclusion. Since we have an extremely large sample size of"
> print("10,000, the continuity correction is completely unnecessary.")
[1] "10,000, the continuity correction is completely unnecessary."
```

References:

- http://imaging.mrc-cbu.cam.ac.uk/statswiki/FAQ/yates